National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

University of Alberta

Evaluating Fairness in Broadcast Bus Networks

by

Vincent Maciejewski

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Masters of Science

Department of Computing Science

Edmonton, Alberta
Fall 1990

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-65117-2

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Vincent Maciejewski

TITLE OF THESIS: Evaluating Fairness in Broadcast Bus Networks

DEGREE: Masters of Science
YEAR THIS DEGREE GRANTED: 1990

Permission is hereby granted to UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis not extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) . . . . . .

Permanent Address:

256 Burton Road,

Edmonton, Alberta,

T6R 1P3, Canada

Date: May 9, 1990

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **Evaluating Fairness in Broadcast Bus Networks** submitted by **Vincent Maciejewski** in partial fulfillment of the requirements for the degree of Masters of Science.

W. Dobosiewicz

P. Gburzynski

C. G. Englefield

U. M. Maydell

P. Rudnicki

Date: April 30, 1997

Abstract

Unfairness is a phenomenon which occurs naturally in fast unidirectional bus networks. Unfair protocols for unidirectional busses have been universally rejected, simply because they are unfair. However, most protocols proposed to date deal with unfairness at the expense of performance. The performance of these protocols quickly deteriorates with increasing a: the ratio of the bus length to packet length. Unfair protocols, and especially the simplest ones, have the advantage that their performance is not sensitive to a under one way traffic. In this work, a survey and a characterization of traffic carried by real networks is given. Then, a model for defining performance and measures of fairness is presented. This model is called the "Points of View Model". It defines performance and fairness from the points of view of the packet, the station, the link, and the user. Simulation experiments are performed to show that fairness looks differently from different points of view, and that fairness measured from the same point of view depends on the traffic type. Under certain conditions, a supposedly unfair protocol can turn out to be completely fair. Further simulation experiments show that fairness depends on the maximum packet length (in unslotted protocols), the load, and a for certain traffic types. Emphasis is placed on file server traffic since this traffic is the one most commonly carried by local area networks. We conclude that fairness should not be evaluated under uniform one–way traffic, and that unfair protocols may be suitable for a variety of applications.

Acknowledgements

I would like to thank my supervisors, Włodek Dobosiewicz and Paweł Gburzyński for guidance and encouragement. In addition, I am indebted to my examining committee for many helpful comments and suggestions, and also to Joanne Parenteau for proofreading the manuscript.

Contents

1 Introduction 1
1.1 The Fairness Problem . . . 1
1.2 Motivation . . . 2
1.3 Purpose . . . 3
1.4 Contents . . . 4
1.5 Literature Review . . . 6

2 Traffic Characteristics 7
2.1 File Server Traffic . . . 7
2.1.1 File Server Design . . . 7
2.1.2 A Model For File Server Traffic . . . 10
2.2 Remote Procedure Call Traffic . . . 12
2.2.1 Traffic Characteristics . . . 12
2.3 Voice Traffic . . . 13
2.3.1 Speech Packetization . . . 14
2.3.2 Speech Reconstruction . . . 15
2.3.3 A Model for Voice Traffic . . . 17
2.4 Video Traffic . . . 18
2.4.1 Video Coding . . . 19
2.4.2 A Model for Video Traffic . . . 20
2.5 Remote Logins . . . 22
2.5.1 Terminal Interaction . . . 22
2.5.2 A Model for Remote Login Traffic . . . 23
2.6 A Classification of Traffic Types . . . 23

3 A Classification of Performance and Fairness Measures 25
3.1 Points of View . . . 25
3.2 A Definition of Fairness . . . 26
3.3 A Definition of Performance . . . 26
3.3.1 A State Transition Model . . . 26
3.3.2 Performance . . . 27
Performance of Systems Characterized by an Event Sequence . . . 28
Performance of Systems Characterized by a Current State Function 28
3.3.3 Performance of Classes of Systems . . . 29
3.4 The Packet Point of View . . . 29
3.4.1 The Model . . . 29
3.4.2 Performance . . . 30
Queuing Delay . . . 30
Acquisition Delay . . . 30
Propagation Delay . . . 30
3.4.3 Fairness . . . 31
Acquisition Fairness . . . 31
Propagation Delay Fairness . . . 31
Queuing Fairness . . . 31
A Representative Measure of Packet Fairness . . . 32
3.5 The Station Point of View . . . 32
3.5.1 The Model . . . 32
3.5.2 Performance . . . 33
Throughput . . . 33

Acquisition Delay . . . 33
3.6 The Link Point of View . . . 33
3.6.1 The Model . . . 33
3.6.2 Performance . . . 34
3.6.3 Fairness . . . 35
3.7 The User Point of View . . . 35
3.7.1 Performance . . . 35
Voice and Video . . . 35
File Server . . . 36
Remote Procedure Calls . . . 37
Remote Logins . . . 37
3.7.2 Fairness . . . 37

4 Simulation Studies in Behavior of Unfair Protocols **38**
4.1 Introduction and Summary of Results . . . 38
4.2 Fairness versus Traffic Type . . . 40
4.2.1 Uniform Traffic . . . 40
4.2.2 File Server Traffic . . . 41
4.2.3 Voice Traffic . . . 45
4.2.4 RPC Traffic . . . 47
4.3 Behavior of Unfair Protocols Under File Server Traffic . . . 48
4.3.1 Single Unidirectional Bus . . . 48
Simulation Parameters . . . 48
LCSMA-CD . . . 49
Variants of LCSMA-CD . . . 50
Effect of Packet Length . . . 51
Comparative Performance . . . 53
Sensitivity to Service Time . . . 57
4.3.2 Dual Unidirectional Bus . . . 59
LCSMA-CD . . . 59
Comparative Performance . . . 61
FASNET . . . 61
LCSMA-CD/S . . . 62
LCSMA-CD . . . 62

5 Summary and Conclusions **63**
5.1 Contents Summary . . . 63
5.1.1 Chapter 2 . . . 63
5.1.2 Chapter 3 . . . 63
5.1.3 Chapter 4 . . . 64
5.2 Conclusions . . . 66
5.3 Possibilities for Further Research . . . 67

A Bus Topologies and Protocols **68**
A.1 Unidirectional Bus Topologies . . . 68
A.2 Protocol Descriptions . . . 69
A.2.1 LCSMA-CD . . . 69
A.2.2 LCSMA-CD/U/P . . . 69
A.2.3 LCSMA-CD/S/P . . . 69
A.2.4 LCSMA-CD2 . . . 70
A.2.5 LCSMA-CDQ . . . 70
A.2.6 LCSMA-CDQ2 . . . 70
A.2.7 p_i-Persistent . . . 70
A.2.8 DQDB . . . 70
A.2.9 Expressnet . . . 71
A.2.10 FASNET . . . 71

List of Tables

2.1 File server traffic simulation parameters. 11
2.2 Voice traffic parameters. 18
2.3 Video traffic parameters. 21
2.4 Remote login traffic parameters. 23

4.5 Effect of packet length on fairness 51
4.6 Average response times for Expressnet and LCSMA-CD with file servers in the least privileged positions. 53

List of Figures

1 NTI voice reconstruction strategy. 16
2 CTI voice reconstruction strategy. 17
3 The data stream model. 22

4 FSM representation of packet behavior 30
5 FSM representation of station behavior. 33
6 FSM representation of a single link cell. 34

7 Performance from the packet point of view under uniformly distributed one-way traffic for LCSMA-CD on the single unidirectional bus. 40
8 Performance from the packet point of view and the user point of view at different stations for single bus unslotted LCSMA-CD under FS traffic with server in the most privileged position. 41
9 Performance from the packet point of view and user point of view for LCSMA-CD on single bus under FS traffic with server in least privileged position. . 42
10 Response time for LCSMA-CD on the single unidirectional bus with FS in most privileged position for different values of a. 43
11 Performance from user point of view for LCSMA-CD on the single unidirectional bus with FS in least privileged position for different values of a. . . . 45
12 Performance from the packet point of view for LCSMA-CD on the single 100 Mbit/s unidirectional bus under voice traffic. 46
13 Acquisition time cumulative distribution function at least privileged station. 46
14 Performance from the packet point of view and user point of view for LCSMA-CD on the single unidirectional bus under RPC traffic. 47
15 Response time for LCSMA-CD on the single unidirectional bus for different values of a under RPC traffic. 48
16 Effect of maximum packet length on response time. 52
17 Response time with 80 stations on a 10 Mbit/s cable (single bus). 55
18 Response time with 80 stations on a 100 Mbit/s cable (single bus). 56
19 Effect of service time on response time in LCSMA-CD. 58
20 Response time with 80 stations, 1 ms service time. 59
21 Response time with 100 stations, no service time, on a 10 Mbit/s cable (dual bus). 61

22 Single unidirectional bus. 68
23 Dual unidirectional bus. 69

CHAPTER 1

Introduction

1.1 The Fairness Problem

Consider the unidirectional bus protocol which operates as follows. A backlogged station senses the carrier and transmits its packet as soon as it detects that the bus is silent. If during transmission, a packet from upstream arrives, the station aborts its transmission without damaging the oncoming packet. It will try to transmit again as soon as the carrier becomes idle. This strategy is obviously unfair to the downstream stations since they must yield to transmissions from upstream. The most upstream station is also the most privileged and it has a near–zero bus acquisition time. This fact has been recognized and methods of dealing with the unfairness have appeared in the literature.

- Demand assignment multiple access (DAMA) protocols are a family of protocols [13] for unidirectional busses where transmissions are scheduled in a round robin manner. Upstream stations transmit before downstream stations, but any station can transmit only one packet in a train of packets. Thus, the acquisition times for each station are the same, and so the protocol is fair. The problem with this approach is that the performance of DAMA protocols is sensitive to a[1]: the ratio of bus length to packet length. This makes DAMA protocols unsuitable for use in high speed networks.

- Access restriction is another method of dealing with unfairness; it is, however, only applicable to slotted protocols[2]. As proposed by Mukherjee and Meditch [24], backlogged station i transmits its packet in an empty slot with probability p_i. The probabilities

[1] The value of a increases as the transmission speed increases.
[2] All protocols studied in this document are summarized in the appendix.

are adjusted such that the network is fair. The problem with this approach is that in order to calculate each p_i, it is necessary to know the current traffic intensity on the entire network, and the bandwidth demands of each station[3].

The research reported here is done under the assumption that the two solutions above may not be suitable to all applications of broadcast bus networks. It is believed that unfair protocols can prove better than fair protocols under real traffic conditions.

1.2 Motivation

The original intent of this work was to study fairness in local area networks for the following reasons.

- Very little research on fairness in broadcast networks has been done before. It is generally assumed that unfair networks are uninteresting. Thus their merits are usually overlooked.
- No satisfactory solution to the fairness problem has been proposed.
- Unfairness is a phenomenon which occurs naturally in fast unidirectional bus networks. Most protocols incorporate mechanisms that make these protocols "artificially" fair. These mechanisms usually trade performance for fairness.
- Some unfair networks are insensitive to a, and are therefore well suited for use with high transmission rates.
- Preliminary simulation studies showed that the unfairness of a network is strongly influenced by the traffic type. Measuring acquisition time at each station on an unfair network under uniform traffic, and the response time under request/reply traffic gave different results. Specifically, the protocol was more fair in the latter case.

Further research in fairness presented some serious obstacles. A network can be defined to be unfair when performance measured at different stations on the bus differs. But

[3] Methods for dynamic adjustment of the p_i's have already been proposed.

the term "performance" is rather loosely defined—it is not clear which single performance measure is appropriate. There exist a large number of performance measures and their definitions vary from publication to publication. It became obvious that before fairness can be studied, it is necessary to define performance.

Different performance measures are appropriate under different traffic types. For example, for voice traffic, it is the number of late voice packets, and for file server traffic, it is the time measured from the moment a request is queued at a client station to the moment the reply is received from the server[4]. However, there exist performance measures which appear to be useful for more than one traffic type. For example, measuring throughput and acquisition time is not unreasonable regardless of the traffic type. In addition, it is not clear what traffic type, and what parameters should be used for the measurement of fairness. Uniform traffic, with Poisson arrivals and one–way transmissions, is a convenient starting point, but the results cannot be extrapolated to real networks where other kinds of traffic prevail. These problems are summarized by the two questions below.

- Which performance measure, or measures, should be chosen for the measurement of fairness?
- What are the simulation models and parameters for traffic found on real networks?

This work does not attempt to answer the first question directly, but rather to provide a framework which could be used to formulate the most appropriate answers. Some assumptions have to be made nevertheless. It is assumed that the traffic type should be as realistic as possible so that the results obtained by simulation can be carried over to real life situations. And it is also assumed that more than one performance measure may be of interest, and that this will depend on the traffic type. Chapter 2, a survey of traffic types, is intended to answer the second question.

1.3 Purpose

The purpose of this work can be summarized in four points.

[4]This is later defined as the *response time*.

- Survey and characterize traffic types most commonly carried by broadcast local area networks, and specify their simulation models and parameters.
- Provide a framework for formal definition and classification of performance measures, define the performance measures most commonly used, and emphasize the ones which appear to be best suited for the measurement of fairness.
- Give examples of situations where unfair protocols outperform fair protocols.
- Perform some preliminary studies on the behavior of unfair protocols under realistic traffic conditions.

These goals are justified as follows. Since the traffic type affects fairness, protocols should be studied under realistic traffic conditions. The second goal is intended to provide a method for selection of appropriate performance measures for the measurement of fairness. The third goal is intended to show that unfair protocols can deliver superior performance as compared to fair protocols. And finally, the last point is intended to expose topics for further research, and to provide some information on factors affecting fairness.

1.4 Contents

The organization of the thesis reflects its purpose. The second chapter is a comprehensive survey of realistic traffic types. The types considered are: voice traffic, video traffic, file server traffic, remote login traffic, and remote procedure call traffic. A simulation models for file server, remote procedure call, and voice traffic are developed. The parameters for each model are given according to measurements given in other publications. No in-house traffic measurements are performed due to practical limitations.

The third chapter presents a method for classification and formal definition of performance measures. Performance is considered from different *points of view*. For example, the network looks differently from the point of view of the packet than from the point of view of the link. Different performance measures are defined according to the viewpoint. From the packet point of view it is possible to define the acquisition time, but this cannot

be done from the point of view of the link. Channel utilization, on the other hand, can be defined from the point of view of the link but not the packet. The user point of view allows for the definition of performance measures which are used by the human user in the evaluation of the utility of the network. Packets, links, and so on, are referred to as network entities. Performance from the point of view of a certain entity is defined using the behavior of that entity. Behavior of network entities is modeled using finite state machines. Formal definitions of performance and fairness are given. The packet, station, link, and user points of view are considered. Among the performance measures defined from these four points of view are the following: queuing delay, acquisition delay, propagation delay, throughput, and link utilization. Performance and fairness from the user point of view are also discussed. Parts of Chapter 3 also appear in [10].

The fourth chapter studies the performance and fairness of a very simple protocol that could be described in the following manner:

- Whenever a station wishing to transmit senses silence, it starts transmitting.
- If a station senses a collision while transmitting, it aborts its transmission. This transmission is to be restarted from the beginning as soon as a period of silence is sensed.

This protocol has no name, although it is obviously generally known. In the classification introduced by Maxemchuk [33], this protocol is labeled local carrier sense multiple access with collision detection and with passive taps, LCSMA-CD/U/P and LCSMA-CD/S/P, the latter being a slotted version of the former. This protocol is suited for unidirectional busses (either a single, or a dual bus). However, it is universally rejected on the grounds that it is obviously unfair, and its potential merits are overlooked. The chapter is an attempt to show that this, and similar protocols, can be better than fair protocols under realistic traffic conditions. A comparison of the unfairness under most of the traffic types mentioned above is made. It is shown that fairness depends on the traffic type, the point of view, the load, and the maximum packet length. In addition, in case of remote procedure call and file server traffic, the fairness is shown to depend on a. The chapter concentrates on one of

the traffic types, namely file server traffic. This traffic is common to local area networks, and it affects the fairness of LCSMA–CD in a counterintuitive way. Remote login and video traffic are not studied, primarily due to time restrictions. The bandwidth required by remote login traffic is very small and is not likely to be a significant factor in most networks. Video traffic, on the other hand, requires very large amount of bandwidth and is not usually carried by local area networks. However, video traffic is expected to be carried by LAN's in the future. The characteristics of fairness under this traffic type are expected to be the same as those for voice and uniform one–way traffic. Parts of Chapter 4 also appear in a separate document [29].

1.5 Literature Review

The body of literature on fairness in broadcast networks is rather small. The first, and only paper devoted primarily to fairness is a short work by Marsan and Gerla [31]. Unfortunately, it only gives a definition of fairness and a short comparison of several unfair protocols. It concludes that fairness is not a problem in local area networks since unfair protocols were not well known at that time. Another noteworthy development is the work on the p_i-persistent protocol by Mukherjee and associates [24, 36, 37, 38]. Fairness is not the primary objective in these studies, but a protocol for unidirectional busses whose fairness can be adjusted is studied. This protocol, and others, are briefly explained in the appendix. Finally, and most importantly, an unfair metropolitan area network appears to be gaining acceptance. This protocol is called the Distributed Queue Dual Bus (DQDB) protocol [9, 12] and it has been proposed for standardization. This protocol operates in essentially the same way as slotted LCSMA–CD on the dual unidirectional bus, except that the maximum packet delay is bounded.

The Points of View classification method is not know to have appeared in previous publications. However, the underlying idea is obviously not new. For example, L. Svobodova considers performance at the user level and at lower levels [55].

CHAPTER 2

Traffic Characteristics

2.1 File Server Traffic

The designs of file servers differ substantially [56]. For example, files can be stored contiguously or in blocks. The number of operations on files that the file server recognizes may vary from as little as three to over a dozen. There are several different ways requests can be acknowledged, and file servers implement atomic transactions differently. Several different caching strategies at the server and at the client can be implemented. The traffic is influenced by all of the above. We discuss file servers in general, and then formulate a simulation model.

2.1.1 File Server Design

Block Transfer and Whole File Transfer

Many file servers such as the SUN Microsystems NFS (Network File System) file server [49], provide access to files on the block level. This is particularly useful in environments where the workstations have virtual memory and are diskless. When a page fault occurs, the file server is requested to supply the page. Blocks or pages which are logically contiguous are not necessarily stored that way on the disk. This strategy prevents fragmentation. If files were stored in their entirety without being split, fragmentation would inevitably result because of unusable holes between files. However, if a file were to be read or written sequentially from the beginning to the end, then this operation could be slower, provided the file is stored in several blocks since more than one disk access is likely to be required.

Statistics show [44] that in a UNIX file system about 70 percent of all file accesses result

in transfer of the whole file. Other studies [39] indicate that as much as 99 percent of all files are less than 64K bytes in size, and that the median file size is only 1K bytes in a UNIX system. Since workstations with 8M of memory are common today, and workstations with 16M bytes of memory will not be uncommon in the future, most accessed files will fit in the main memory in their entirety. In addition, paging will not be as frequent. If workstations become equipped with local disk drives, all necessary process swapping may be performed locally. Given that most files are accessed as a whole, anyway, the advantages of contiguous file storage on the file server and whole file transfer are obvious. It has been reported in [21] that whole file transfer decreases the load on the file server and the network. Several file servers which use this scheme have already been implemented including the Andrew ITC file server [21], and the Bullet file server [58] for the Amoeba distributed operating system [59]. It has been reported that the Bullet file server is about three times faster than the NFS (Network File System) file server.

The drawback of the whole–file approach is that at the moment it appears to be unsuitable for append operations, and databases. To solve the first problem the Amoeba operating system implements separate servers to handle logging and append operations. Data bases can be split into a collection of small files. Special data base servers are also being developed.

File Caching

Most file servers implement a caching mechanism to decrease the number of disk accesses and thereby improve the average transfer rate. For example, the file server might check to see if the requested file is in the cache before reading it from the disk. This is not taken into account in the model presented below.

Acknowledgement Procedure

The communication medium is never entirely reliable. To ensure that requests from the client reach the server, acknowledgements and timeouts are used. In existing file servers this has been implemented in two ways [56]:

- *Requests/Response:* The client sends a request and later expects a response. If the response does not arrive due to a network error or some other error, then the client retransmits the request. The server must be able to handle duplicate requests. The ones which have already been fulfilled may be especially troublesome.

- *Requests/Response/Ack:* In this case the server, having received the acknowledgement from the client, can be certain that a duplicate request will not arrive, and may update its data structures accordingly.

The ability to deal with duplicate requests is a function which can complicate the design. The alternative is to make operations on files idempotent: repeated operations on the same file have a collective effect equivalent to the execution of a single operation. If all operations are idempotent then acknowledgements are not necessary.

All requests in the Bullet file server are idempotent because it provides only three requests: read, create, delete. The only way to update a file is to create a new version of the file and then delete the old version. This solution appears to be quite appealing and we anticipate that it will be used in other file servers of this type.

Connection-Oriented and Connectionless Protocols

Connection oriented protocols construct virtual circuits between the communicating entities. Such protocols have three distinct phases: connection establishment, transfer, and connection release. In connectionless protocols, flow and error control is not observed. Data packets have no relation to other packets sent between the sender and the receiver, and error control is implemented by the higher layers. As explained earlier, in file transfer protocols this is often handled by retransmission of the requests. It is generally agreed that connectionless protocols at low levels are appropriate as long as the packet loss rate is low. However, if packets are often lost due to buffer overflow and other errors, a connection-oriented protocol for file transfer may be more appropriate [34].

2.1.2 A Model For File Server Traffic

Required Bandwidth

The bandwidth required due to file transfer will differ significantly from installation to installation. However, several measurements have been made in UNIX programming environments. In [44] the amount of data transferred by an active user was measured to be between 300 and 600 bytes/s and in [27] the bandwidth was measured to be approximately 4000 bytes/s. In the former study, an active user was defined as one who has caused some I/O in an interval of 10 s. In the latter, software developers were told to "work hard" for 30 minute intervals during which they were supervised and measured (cf. [27]). In both cases, however, access to files was on the level of blocks. When whole-file transfer is done, it is likely that the bandwidth requirements will be different.

File Transfer

The data unit transferred is the file. The median file size is about 1K bytes [39]. Unfortunately, no data has been found on the frequency of access. We must therefore be content with the following conjecture. We assume that short files are accessed more frequently than long files, and that very long files are accessed very rarely. Given this assumption, it is not unreasonable to postulate that the average amount of data transferred per file access is equal to the median file size. Since the bandwidth is 4000 bytes/s, we must have 4 file transfers per second. This number is application and system dependent and it should only be used as a guideline. In Chapter 4 we use 10 file transfers per second.

We further assume that the ratio of read[1] operations to create operations is 4:1[2]. A read operation involves two steps. First, the client station sends a read request to the file server. The file server then sends the whole file back to the client station. The create operation involves two steps also. The client station sends the whole file to the file server. In the second step, the file servers sends an acknowledgement back to the station.

Another assumption we are going to make is that the transferred file length distribution

[1] This also includes the delete operations.

[2] This ratio is somewhat arbitrary. The ratio of 3:1 was used in [27].

parameter	*value*	*distribution*
Mean Transferred File Size	1000	exponential
Requests	Read & Create	
Read/Create Requests	4:1	
Frequency of Request	4 per sec	
Data Unit	Whole File	
Session Duration	30 requests	exponential

Table 2.1: File Server Traffic Simulation Parameters

is exponential with the mean of 1000 bytes. The transferred file length is the stored file length weighted by the frequency of access.

The parameters are summarized in Table 2.1.

Implementation

There are two kinds of stations: file servers and client stations. Each file server contains a server process which handles the incoming requests from the client stations. The client stations contain a client process which generates requests which are transmitted to the file servers. The file servers have infinite request queues. When a request cannot be attended by the file server immediately, it is put in the queue. The pseudocode for the client process is given below.

```
while (1) {
  req = generate_request ();
  send (req);
  if (req == CREATE)
    wait_for_acknowledgement ();
  else if (req == READ)
    wait_for_file ();
  sleep (generate_req_interarrival_time ());
}
```

The functions **wait_for_acknowledgement** and **wait_for_file** block the process until either the acknowledgement or the file is received, respectively. The function **sleep** suspends the process for a period of time equal to its argument. The function **send** is non-blocking: it queues the message given as its argument for transmission. The pseudocode for the server process is given below.

```
while (1) {
  req = wait_for_request ();
  sleep (generate_request_service_time ());
  if (req == READ)
    send (generate_file ());
  else if (req == CREATE)
    send (generate_acknowledgement ());
}
```

As expected, the function **wait_for_request** blocks the process until a request is received.

2.2 Remote Procedure Call Traffic

2.2.1 Traffic Characteristics

The remote procedure call is a widely accepted mechanism for interprocess communication, especially for processes placed on physically different machines. In this section a distinction is made between the kind of RPC (remote procedure call) traffic resulting from communication with the file server, and the kind of traffic resulting from interprocess communication. The communicating processes are assumed to reside on physically distinct machines.

The RPC mechanism can be described as follows [4]. A client process transmits a request to a remotely located server, and blocks until a packet containing the result arrives. When the packet is received, control is given back to the calling process. The frequency and size of RPC packets is obviously dependent on the network operating system. Measurements for the Taos (the operating system for the Firefly workstation) and NFS operating systems are reported in [2]. On a single Firefly workstation 18366 network RPC's were made in a five hour period, which equals roughly 1 RPC/s. Since the workstation was connected to a remote file server, a large portion of those RPC's could have been addressed to the file server. The measurements reported did not report the proportion of packets sent to the server.

It is probable, that the reason for the low frequency of RPC's is due to the fact that network operating systems are designed in a way which minimizes the number of remote

calls. Local calls are preferred for reasons of efficiency. However, it is anticipated that the situation will change with the advent of very fast local area networks whose bandwidth is comparable to memory bandwidth.

Other measurements [50] show that the time to service an RPC is on the order of several milliseconds for most network operating systems in use today.

The Model

The model for RPC traffic differs little from the model for file server traffic. The file server model differs in that the client stations contain the client process and server stations contain the server process only. In case of RPC traffic, each station which participates in this traffic contains both processes.

It is not possible to give guidelines for the choice of parameters for simulation experiments because no pertinent publications were found. For the purposes of this work we will use similar parameters as for file server traffic. There will be only one kind of request, and the length of the reply will be 1000 bytes, exponentially distributed. The frequency of requests will be adjusted, depending on the number of stations on the network.

2.3 Voice Traffic

Human speech contains a significant amount of silence time between consecutive utterances of words, and even syllables. In conversation, speakers normally take turns at speaking, and there are additional silence periods when speakers switch. In a conversation where there are two participants, we would expect that the average speaker will be active for less than half of the time. Empirical studies [5, 6] show that in telephone conversations the average speaker is active for 40 percent of the time. Given that there is a full-duplex channel connecting the two speakers, that channel will be idle for 20 percent of the time. If there are two half-duplex channels, then each channel will be idle for 60 percent of the time. Speech, band-limited to 8 kHz, can be digitized to 64, 32, and 16 kbits/s. Given a transmission rate of 100 Mbits/s, one second of speech can be transmitted in 0.16 ms.

Suppressing the silence periods an additional saving can be made. This allows for statistical multiplexing of a number of conversations on a shared medium.

2.3.1 Speech Packetization

On a local area network, speech is transmitted in packets. The stream of packets is not periodic because silence periods are suppressed. The distributions of talkspurt and silence intervals have been found to be exponential with means of 1.2 s, and 1.8 s respectively [14]. Since packets are transmitted individually, each packet can experience a different delay.

Hangover Interval and Delay Variability

The speech detector is the device which informs the speech encoder whether or not a speech signal is present and whether packets should be produced. An important parameter of the speech detector is the *hangover interval.* This is the time period during which the speech detector conjectures that speech is present even though speech input power has dropped below a certain threshold. Low values of hangover give more silence periods, but many of the silence periods are also very short. This results in silence periods between individual utterances. Longer hangover intervals result in longer talkspurts, and therefore the silence periods occur only between words, phrases, and sentences. Little research appears to have been done in this area, but studies indicate that if the hangover interval is set to 150–300 ms, then the quality of speech (English narratives) is not affected by the variability of the silence periods, as long as the average variability does not exceed 150–300 ms [18]. Excessive variability in silence periods can affect the meaning of what is being said. However, if the hangover time is set to less than 100 ms, the quality of speech deteriorates rapidly for average variability in silence periods of more than 50 ms. The average variability of 150 ms on a 100 Mbit/s network is equal to the transmission time of 15 Mbits. It is therefore unlikely that the variability could exceed the limit of acceptability on a fast network.

Maximum Packet Delay

There exists a subjective upper bound on the packet delay. For good quality reproduction it has been proposed that this upper bound should be between 100-200 ms [8, 11], however, in the public telephone network this bound is set to 600 ms [26].

Choice of Packet Length

The packet length should be kept less than some allowable value so that the delays of other packets waiting for the transmission of a long voice packet do not become excessive. Also, longer packets add to the variability in packet delay. This, however, is not a concern for voice systems based on LAN's because 1 s of speech takes 0.16 ms to transmit on a 100 Mbit/s network, and 1.6 ms on the standard Ethernet. Therefore, the effect of packet length on the delay is rather small in LAN's but is a concern in slower networks [35, 54]. It is obvious, nevertheless, that packets cannot contain too much speech. If the maximum allowable packet delay is 200 ms and an individual packet contains more than that, then the maximum delay will be exceeded. A general agreement in the literature exists that packets containing about 10-50ms [7, 54, 61] give good speech quality.

2.3.2 Speech Reconstruction

The goal of the reconstruction process is to reproduce the silent periods and talkspurts registered by the packet voice transmitter (PVT). The choice of reconstruction strategy can affect the number of lost packets and therefore may be important in determining the maximum number of conversations a network can carry. There are three commonly used methods of reconstruction [1, 54]:

Null Timing Information (NTI) Strategy

The NTI strategy is shown in Figure 1. The packet voice receiver (PVR) adds a fixed amount of delay d_0 to the first packet (e.g. 40 ms) in the talkspurt. Subsequent packets in the talkspurt are played out in the sequence of arrival. The packets are expected to have

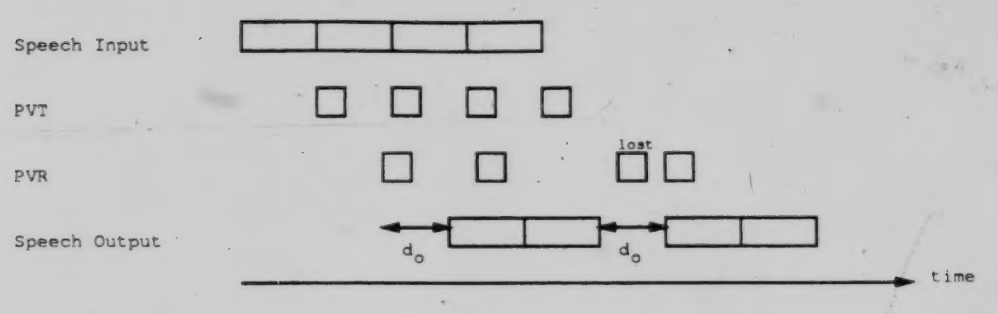


Figure 1: NTI voice reconstruction strategy.

arrived before their scheduled play–out time[3]. If a packet arrives after its scheduled play–out time it is considered "lost". The lost packet and packets following it in the talkspurt could still be played out (as shown). If sequence numbers are transmitted, then the lost packet could be played out immediately upon arrival.

Complete Timing Information (CTI) Strategy

The CTI strategy is shown in Figure 2. The packets are time-stamped by the PVT. If the delay of the first packet in the talkspurt through the network is d_N, and $d_N < d_0$ for some d_0, then the packet is additionally delayed by $d_0 - d_N$. Any packet in the talkspurt whose delay $d_N > d_0$ is considered "lost". This technique reconstructs the silence intervals more accurately than NTI but requires that timing information be transmitted and that the clocks be synchronized.

NTI–CTI Mix Strategy

If the delay of the first packet in the talkspurt through the network is d_N, and $d_N < d_0$ for some d_0, then the CTI strategy is used. If $d_N \geq d_0$, then the first packet is played out immediately and the subsequent packets are handled as in the NTI strategy. Packets which arrive after their scheduled play–out time are considered "lost".

[3]The packet must be ready for play–out at its scheduled play–out time if continuity of sound is to be maintained.

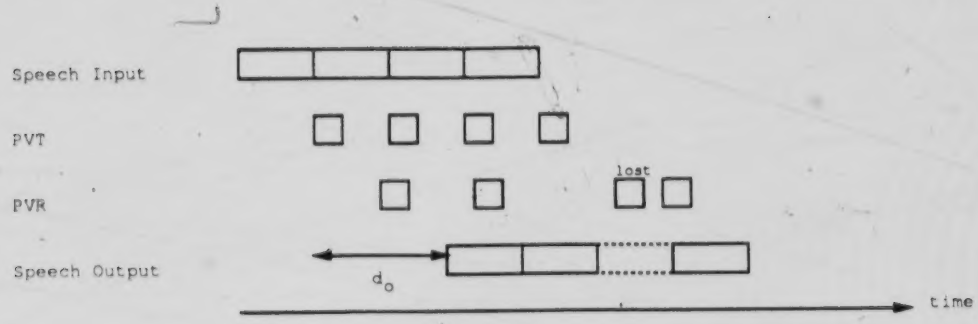


Figure 2: CTI voice reconstruction strategy.

2.3.3 A Model for Voice Traffic

We choose NTI as the reconstruction strategy because it is the one most sensitive to delays, and the number of lost packets should be similar to that obtained with CTI and higher than that obtained with NTI-CTI. It is intended that the NTI strategy be indicative of the worst-case scenario. It is also possible not to use any reconstruction strategy but in that case it is likely to exaggerate the number of lost packets.

We will not be concerned with the variability in packet delay. Instead we assume a hangover time which assures that variability of up to 150 ms is acceptable to the user. Given a delay in NTI of 40 ms, there could be many lost packets if the variability were ever to approach 150 ms. For this reason, we believe, it will suffice to measure the ratio of lost packets.

As the measure of packet loss in a given network for voice transmission we use the formula

$$\zeta = \frac{\text{number of packets lost}}{\text{number of packets received}} \tag{2.1}$$

In simulation experiments we will investigate the relationship between ζ and the number of stations on the network. Values of loss of up to 1 percent [17] have been found to be acceptable to the listener.

We assume that speakers exchange talkspurts. When one of the speakers is in the talk state, then the other is silent. We also assume that both speakers never talk at the same time. In real life situations this is not true, and a more realistic model should take this

parameter	*value*	*distribution*
Speech Bandwidth	8 kHz	
Transmission Bandwidth	64 kbits/s	
Mean Talkspurt Duration	1.2 s	exponential
Mean Silence Duration	1.8 s	exponential
Hangover Interval	200 ms	
Packet Speech Content	50 ms	
Reconstruction Strategy	NTI	
Delay (d_0)	40 ms	
Maximum Lost Packets (ζ)	1 %	

Table 2.2: Voice Traffic Parameters

possibility into account. However, it rarely[4] happens that both speakers are active at the same time and therefore the simulation results should not be significantly affected. Each speaker executes the following code.

```
while (1) {

  /* talkspurt */
  sleep (inter_talkspurt_gap ());
  n = number_of_packets_in_talkspurt ();
  for (i = 0; i < n; i++) {
    transmit_packet ();
    sleep  (inter_packet_gap);
  }

  /* silence */
  packet = receive_packet ();
  while (!is_last_packet (packet)) {
    sleep (packet_play_out_time);
    packet = receive_packet ();
  }

}
```

Voice traffic parameter values are summarized in Table 2.2.

2.4 Video Traffic

Digital pictures are created by sampling in time a scene to produce a sequence of frames. The frames are then sampled in two directions to give a matrix of picture elements (pels). In the standard NTSC television, 30 frames per second are displayed. The number of

[4] About 5 percent of the time [47].

pels will depend on the desired picture quality. Video coding [41, 42], takes advantage of the correlations that exist between pels in a given frame (intra–frame coding) and the correlations between successive frames (inter–frame coding).

2.4.1 Video Coding

Coding Techniques

The most straightforward method of coding is pulse code modulation (PCM.) This is simply a time and amplitude–discrete representation of the signal sampled at the Nyquist rate. PCM is rarely used as a coding method all by itself since it results in large amounts of data. Even a small 256 by 256 pel picture with 128 gray levels transmitted at 30 times per second would require 14 Mbits/s, and black and white television sampled at 8 MHz with 128 digitization levels requires 56 Mbits/s. Therefore, PCM is usually employed as the first step in the coding process.

In *predictive* coding schemes, often referred to as differential PCM (DPCM), a prediction of the sample to be encoded is made from previous samples and only the error of prediction is transmitted. *Conditional replenishment* is an inter–frame variation of this technique. Here, if the error is below a certain threshold then it is not transmitted. This technique is especially effective for scenes which include little motion such as videotelephone–type scenes where empirical data [20] shows that only about 9 percent of samples change by more than 1.5 percent from frame to frame.

Coding techniques now allow for broadcast quality video to be transmitted at the rate of 15–30 Mbits/s [40, 62], and teleconferencing quality video at the rate of 0.3–2 Mbits/s [48] and even as low as 56Kbits/s (0.08 bits/pel) [52].

Variable and Fixed Rate Coding

The bit rate generated by a video codec is inherently nonuniform. Take for example the conditional replenishment strategy. The bit rate generated during scenes which include a significant amount of motion will be much higher than the bit rate generated while transmit-

ting a scene showing a single person listening. In circuit switched networks, these variations are smoothed out to produce a constant-rate source. This can be done using a buffer and a feedback mechanism to control the bit rate when the buffer becomes full. This procedure results in variable quality video. In packet switched networks, variable rate coding offers several advantages [60]. The quality of the video signal is constant. The complexity of the codec may be reduced since the feedback mechanism and the buffer are no longer needed. The bandwidth is utilized more efficiently because a variable rate codec produces bits at an average rate which is lower than the constant rate of a fixed rate codec. Multiplexing several sources on one communication channel offers a statistical gain of about 2 [19, 25].

Variable rate coders used in packet switched networks allow the picture to be coded *hierarchically* [60]. Hierarchical coding, sometimes referred to as layered or embedded coding, splits the signal into several subsignals of different importance. For example, the picture could be separated into bit-planes [3] where the most important subsignal would be the most significant bit-plane. Another example is subband analysis [23], where the video signal frequency spectrum is split in three dimensions: vertically, horizontally, and temporarily. Hierarchical coding allows session quality to be selected by the user before the initiation of the session. An important application of hierarchical coding is the graceful degradation of quality during network congestion. That is, the most important information would be transmitted first, and if there are late packets then the frame may still be reproduced without them if the information contained in the late packets is not significant.

No published data on the maximum acceptable number of lost packets has been found, but we assume, pessimistically, that no more than 1% can be lost.

2.4.2 A Model for Video Traffic

Characteristics of Variable Rate Video Sources

As explained earlier, variable rate video sources are bursty. The exact nature of the burstiness depends on the coding algorithm used. It has been observed that video sources can be

parameter	*value*
Transmission Bandwidth	1.5 Mbits/s
Frame Frequency	30 s^{-1}
A	0.8781
B	0.1108
η	55000
Receiver Buffer Size	4 frames
Maximum Lost Packets	1 %

Table 2.3: Video Traffic Parameters

modeled by the fist order autoregressive Markov process which can be generated by

$$\lambda(n) = A\lambda(n-1) + Bw(n)$$

where $\lambda(n)$ is the number of bits generated by encoding the n'th frame, and where $w(n)$ is a sequence of independent random variables [30]. The sequence $w(n)$ has the Gaussian distribution with variance 1 and mean η. Assuming that $|A| < 1$ ensures that the process achieves steady-state. The steady-state average and discrete autocovariance are given by

$$E[\lambda(n)] = \frac{B}{1-A}\eta,$$

and

$$C(n) = \frac{B^2}{1-A^2}A^n, n \geq 0,$$

respectively [45]. The applicability of the autoregressive model has been supported and several variations have been introduced in [43, 51, 52].

The number of bits generated per frame depends on the coding technique and on the amount of motion in the scene. Empirical data for two head–and–shoulders scenes, one showing a person talking and another showing a person listening are given in [43]. The scene showing a person talking involves more motion. Three different coding techniques were used. For simplicity, however, we will use the data given in [30] where the average of the two scenes was taken. The coding method was a variation of conditional replenishment, and frames were generated 30 times per second. The values of A and B were found to be approximately 0.8781 and 0.1108, respectively.

We assume that the mean bandwidth required by a transmitter is 1.5 Mbits/s, and that

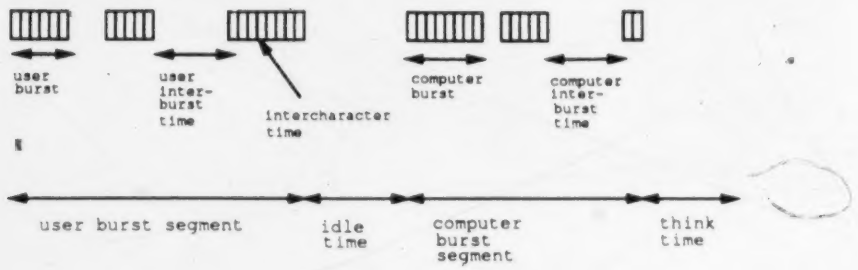


Figure 3: The data stream model.

frames are generated 30 times per second. Therefore, $E[\lambda(n)] = 50$ Kbits, and $\eta = 55000$ bits. The parameters are summarized in Table 2.3.

2.5 Remote Logins

2.5.1 Terminal Interaction

Remote logins do not differ significantly from normal terminal interaction. Since every character sent to the host is echoed, the number of characters transmitted from the remote host back to the terminal can be no smaller than the number of characters sent to the host. Actually, the ratio of characters received to characters transmitted by the terminal is much larger than 1 to 1. This ratio has been observed to vary from over 35 to 1 for technical sessions, to 3 to 1 for word processing sessions [32].

It is apparent that during normal interactive work the user does not type continuously, but rather, sometimes awaits the response from the host, and sometimes just thinks [15, 22]. The time intervals during which the user sends characters to the host are called *user burst segments*, and the time intervals during which the host sends characters to the terminal without the user typing are called *computer burst segments*. The segments are collections of bursts. If a user burst follows a computer burst, then the two are separated by the *think time*. If a computer burst follows a user burst, then the two are separated by the *idle time*. The short idle period between characters is called *intercharacter time*. This model is referred to as the stream model, and is shown in Figure 3.

The distributions of the time periods mentioned above have been found to correspond

parameter	*mean*	*std*	*distribution*
Call Length	81 burst segments	31	geometric
Think Time	5.0 s	3.1	gamma
User Interburst Time	1.5 s	0.83 s	gamma
Computer Interburst Time	13 s	21	gamma
User Burst Segment Length	11 burst	2.8	gamma
Computer Burst Segment Length	2.9 bursts	2.4	gamma
User Burst Length	1.1 chars	0.1	gamma
Computer Burst Length	41	24	gamma
User Intercharacter Time	0		
Computer Intercharacter Time	0		

Table 2.4: Remote Login Traffic Parameters

to well known statistical distributions [15, 46]. The results are summarized in the Table 2.4.

2.5.2 A Model for Remote Login Traffic

In a UNIX system, every time the user strikes a key at the keyboard, a character is transmitted to a remote host in a separate packet. The host echoes each character by transmitting it back. The carriage return indicates the end of the user segment. The carriage return is echoed immediately, and after the idle time period elapses the host sends back its response. This response consists of several computer bursts. Each computer burst is transmitted in a separate packet. The last computer burst is specially marked to indicate the end of the computer segment. After the think time elapses, the user segment begins again.

The statistics of how many characters are transmitted by the host back to the user are not so important because most responses will be quite small compared to the entire packet size. The length of the preamble, header, trailer, and the CRC, will be often larger than the entire response from the host. A more significant parameter is the number of bursts in a computer segment.

2.6 A Classification of Traffic Types

There are some similarities between the different traffic types. For example, file server (FS) traffic, can be thought of as being a special kind of remote procedure call (RPC) traffic.

In this section, common characteristics of the traffic types discussed are abstracted. A classification is given based on those characteristics.

- **Request Reply Traffic.** In this type of traffic, a station makes a request which it sends to a server station, and then expects a reply. No other requests to the same server are generated until the reply arrives. The three traffic types of this kind are remote logins, file server traffic, and remote procedure calls.

- **One–Way Traffic.** In this kind of traffic the receivers of packets are irrelevant. Even though packets may contain an address, it is irrelevant whether the packets are received or not. The traffic which falls into this category is video traffic. Even though video packets are addressed to a specific station, the continuity and amount of information transmitted does not depend on the receiver.

- **Hard Time–Constraint Traffic.** Here, there exists a deadline for the time in which the packet must reach the receiver. The two traffic types which fall into this category are video traffic and voice traffic. If the packet delay is excessive, then the packet is considered lost, however, the packet delay is irrelevant as long it is below the maximum allowed.

- **Soft Time–Constraint Traffic.** In this class of traffic, there are no deadlines for packet delays. A maximum packet delay can be set as a design parameter, but if this maximum is exceeded nothing out of the ordinary will happen. Remote login, FS, and RPC traffic all fall in this category.

- **Synchronous Traffic.** Traffic in this category is distinguished by the fact that packets are generated at equal intervals. Video traffic is the obvious candidate. Voice traffic should also be placed here because packets are generated at equal intervals for a very long period of time followed by a period of silence.

- **Asynchronous Traffic.** Packets are generated at random intervals. Remote logins, FS, and RPC traffic fall into this category.

CHAPTER 3

A Classification of Performance and Fairness Measures

3.1 Points of View

We consider the network performance and fairness from different *points of view.* For example, the network "looks" different, and the measures of performance are different, from the point of view of the packet than from the point of view of the link. The acquisition time is something we can measure by observing the behavior of the packet. However, we cannot measure it by observing the behavior of the link or the user. But if we observe the behavior of the link we can measure the average link utilization. This line of reasoning leads us to propose a method of classification of network performance measures: "Performance is always measured from a certain point of view." The number of points of view is not limited. There are, however, four points of view which correspond to easily identifiable network entities. We shall consider performance from the point of view of the following entities: the packet, the station, the link, and the user. Of course, there may be other entities, such as the network interface unit, but we will limit the discussion to these four. In this chapter, we find that by considering the common points of view we can capture the performance measures used most often.

Shortly, we will formalize the concepts introduced here. We shall define the meaning of "point of view", "performance measure", and "fairness".

3.2 A Definition of Fairness

Intuitively, a local area network is fair if all stations in the same class receive an equal share of network resources. The bigger the share of resources that a station uses, the better the performance of the station. Thus, if all performance measurements taken at each station are equal, then each station has equal access to the available resources, and the network is fair. From this perspective it appears that the measurement of fairness is a measurement of how much performance varies from station to station.

Proposition 1 *A local area network is fair if and only if the performance measured at any station in the network is equal on the average to the performance measured at any other station.*

If a network is unfair, then there are differences in performance measured at different stations.

Definition 1 (Fairness) *Fairness is given by*

$$F = \frac{min(P)}{max(P)}$$

where P is the set of values of a given performance measure for all stations in the network.

3.3 A Definition of Performance

3.3.1 A State Transition Model

We model the behavior of network entities using state transition systems. Consider the system modeling the behavior of the packet. The transitions in this system correspond to real events such as the beginning or end of transmission. The states of the system express what is happening to the packet at that moment. For example, the packet may be: enqueued, waiting to be transmitted, or being transmitted.

We shall be concerned only with entities whose behavior can be modeled by systems with a finite number of states. A system with a finite number of states can partially be

described by a finite state machine. This description is partial because it says nothing about the time of occurrence of the transitions in the system.

Imagine that we have an observer whose sole task is to record events as they occur in some system. That is, whenever a transition occurs, the observer records the name of the transition and the time the transition occurred. We refer to the pair (*transition name*, *time of occurrence*) as an *event*. We call the sequence recorded by the observer the *event sequence* of the system. If the system has a final state, which it eventually never leaves, then the event sequence will be finite, otherwise it will be infinite. The event sequence completely characterizes a system for the purpose of measuring performance. Note that this characterization places no restrictions on the number of states in the system.

There exists an alternative characterization of a system. A system can be characterized by the current state function. Assume that time is expressed as a positive real number. The current state function $\Phi : \Re^+ \rightarrow S$, where S is the set of states of the system, maps positive real numbers into the states of the system.

3.3.2 Performance

Performance from the point of view of the packet can be measured by collecting appropriate statistics about certain events in the behavior of the packet. Observing the state transition system representing the behavior of the packet we are actually observing a subsystem of the system representing the local area network. The system representing the entire network we call the topmost system. The events which occur in the packet system also occur in the topmost system. System Σ_1 is a subsystem of system Σ_2 if and only if there exists a partial mapping $M : B_2 \rightarrow B_1$ of the transitions occurring in B_2, (the behavior of Σ_2), into the transitions occurring in B_1, (the behavior of Σ_1).

Definition 2 (Point of View) *An observer observes a system from a point of view if it observes a subsystem of that system.*

Performance of a network is measured from a point of view if it is measured by observing the behavior of a subsystem of the topmost system representing that network.

Given an event sequence, performance can be measured by measuring the time intervals between occurrences of a certain transition or the time interval between two transitions.

Performance of Systems Characterized by an Event Sequence

Definition 3 (Interval Performance Measure) *Given a system, Σ_1, with the transition sequence $(\alpha_1, t_1, \alpha_2, t_2, \alpha_3)^*$ where $\alpha_1, \alpha_2, \alpha_3$ are transition sequences not including t_1 and t_2, and the subsystem of Σ_1 with the transition sequence $(t_1, t_2)^*$, and the event sequence $\{(t_1, \tau_1), (t_2, \tau_2), (t_1, \tau_3), (t_2, \tau_4), \ldots\}$, the performance measure $\Psi(t_1, t_2)$ is the renewal process with the set of renewal periods $\{\tau_2 - \tau_1, \tau_4 - \tau_3, \ldots\}$. The performance measure $\Psi(t_1)$ is the renewal process with the set of renewal periods $\{\tau_3 - \tau_1, \tau_7 - \tau_5, \ldots\}$.*

For some performance measures only the last or first occurrence of a certain transition is of interest. For example, in the system modeling the packet, we may be interested in the time interval between the last occurrence of the transition *transmission begins* and the occurrence of the transition *transmission ends*. This interval is the transmission time. Consider a system, Σ_1, with the transition sequence $(\alpha_1, t_1, \alpha_2)^*, t_1, \alpha_3, t_2$. To form the required renewal process, we obtain the subsystem Σ_2 with the transition sequence $(\alpha_1, t_1, \alpha_2)^*, t_3, \alpha_3, t_2$ by mapping the last t_1 transition of Σ_1 into t_3 of Σ_2. We assume that $\alpha_1, \alpha_2, \alpha_3$ do not contain t_1, t_2, and t_3. Next, we map Σ_2 into the system Σ_3 with the transition sequence (t_3, t_2). We use superscripts to refer to the last and first occurrences of a transition, and subscripts to indicate membership. Thus $t^L_{1_{\Sigma_1}}$, the last occurrence of transition t_1 in Σ_1, occurs at the same time as $t_{3_{\Sigma_2}}$, the last occurrence of transition t_3 in Σ_2. Similarly, $t^F_{1_{\Sigma_1}}$ refers to the first occurrence of t_1 in Σ_1. We see that $\Psi(t^L_{1_{\Sigma_1}}, t_{2_{\Sigma_1}}) = \Psi(t_{1_{\Sigma_3}}, t_{2_{\Sigma_3}})$.

This interval is the acquisition time for only one packet. Normally, the average acquisition time for many packets is of interest. Section 3.3.3 deals with this issue.

Performance of Systems Characterized by a Current State Function

Performance of systems characterized by the current state function is defined as follows:

Definition 4 (Probability Performance Measure) *Given a system, Σ_1, with the set of states S, the performance measure $\Psi(r)$, where $r \in S$, is the probability of finding the system in the state r.*

3.3.3 Performance of Classes of Systems

Some systems may have more than one instance. For example, during a simulation experiment, there may be thousands of packets sent between stations. There is usually more than one station in the network, and there may be more than one link. The set of all instances of a system is called a *class*. For classes of systems characterized by event sequences we propose the following definition of performance:

Definition 5 (Interval Performance Measure of a Class) *Given a class of systems with the set of instances I, the performance measures $\Psi_c(t)$ and $\Psi_c(t,u)$, where t and u are transitions, are the renewal processes with the set of renewal periods $\bigcup_{i \in I} \Psi_i(t)$ and $\bigcup_{i \in I} \Psi_i(t,u)$, respectively.*

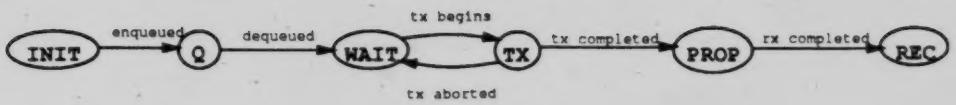
For classes of systems characterized by the current state function we propose the following definition.

Definition 6 (Probability Performance Measure of a Class) *Given a class of systems with the set of instances I, the performance measure Ψ_c is the expectancy of the same performance measure Ψ for each $i \in I$.*

3.4 The Packet Point of View

3.4.1 The Model

The FSM (finite state machine) representing the behavior of the packet is shown in Figure 4. The initial state is the state `INIT`. The packet remains in that state until it is created and enqueued, then it moves to the state `Q`. The packet awaits transmission, in state `WAIT`. It moves to state `TX` after the first bit in the packet is transmitted. The station may choose



. Figure 4: FSM Representation of Packet Behavior

to abort the transmission in which case the packet will reenter the **WAIT** state. Otherwise the transmission is successful and the packet will enter state **PROP** after the last bit in the packet is transmitted. The packet is propagating while in this state. The packet enters the state **REC** after the last bit is received by the receiver station.

The model of behavior of the packet is not cyclic; real packets normally have a finite lifetime. The packet is created, transmitted, received, and eventually it ceases to exist as a packet even though the information which had been contained in it may still exist in some other form. Usually, there are many packets transmitted in a simulation experiment. Therefore, the performance measures defined below refer to the packet class rather than an individual packet system.

3.4.2 Performance

Queuing Delay

Queuing delay is the time the packet spends in the queue at the transmitter station. This is defined as $\Psi_c(enqueued, dequeued)$.

Acquisition Delay

This is the time the packet must wait from the moment it is removed from the queue to the moment its *successful* transmission begins. This is $\Psi_c(dequeued, tx\ begins^L)$.

Propagation Delay

This is the time taken for a signal to travel the distance from the transmitter to the receiver. This is defined as $\Psi_c(tx\ completed, rx\ completed)$.

3.4.3 Fairness

Acquisition Fairness

The acquisition time at certain stations may be greater than at other stations for reasons strictly dependent on the protocol. However, if the traffic intensity is low, even though a given station may be at a disadvantaged position, the difference in the acquisition time may not be significant. Under high traffic intensity, the less privileged stations will have to wait longer. Acquisition unfairness usually increases with increasing load.

Propagation Delay Fairness

The unfairness due to propagation delay is caused by the inequalities in average distances between stations. Consider a network with the bus geometry. A station positioned at an end of the bus will have longer propagation delays than stations near the center if all stations are equally likely to be chosen as receivers of any packet. This kind of unfairness is caused by the topology of the network only, and does not depend on the traffic level in contrast to acquisition fairness. Therefore, propagation delay unfairness is not useful for comparing networks.

Queuing Fairness

The unfairness reflects the fact that some stations have queue lengths which on the average are longer than those in other stations. Packets generated at stations with large average queue lengths will experience longer delays. Longer queues can be caused by either a larger number of messages generated by the station, or acquisition unfairness. But since we have assumed that the bandwidth requirements of each station are equal, longer queues at certain stations must be caused by acquisition unfairness. Since the queue lengths are caused by acquisition unfairness it seems that acquisition fairness is a purer measure of fairness, and makes the notion of queuing fairness redundant here.

A Representative Measure of Packet Fairness

Propagation delay unfairness is not useful for comparing networks because any two networks with the same topology will have equal propagation delay unfairness. A protocol which is queuing unfair is also acquisition unfair[1]. Therefore, measuring both queuing and acquisition fairness appears to be superfluous. Therefore, we suggest that acquisition fairness be used as the representative of packet fairness.

3.5 The Station Point of View

3.5.1 The Model

The station generates, transmits, and receives packets. We shall consider the system representing the behavior of the station from the point of view of the transmitter as shown in Figure 5.

The FSM model of the behavior of the transmitter comprises three states. When the transmitter is in the IDLE state, it has no packets to transmit. In the WAIT state the transmitter has a packet to transmit and waits for the channel to become idle, for the arrival of the token, or for whatever else may be required by the protocol. As soon as it becomes clear that the transmitter can transmit, it enters the state TX and transmits the packet. Two events may occur while the station is transmitting. In certain protocols, transmission may have to be aborted and the transmitter will move back to state WAIT. Otherwise, the transfer is successful and the transmitter goes back to state IDLE.

A more detailed model can be used to illustrate the behavior of the station on the level of individual bits. Such a model might be useful if the station buffer capacity is measured in bits, and if we are interested in the throughput in bits[2]. The performance measures defined below apply both to classes and individual station systems because the behavior of the station is cyclic.

[1] Under the assumption of equal load for all stations.
[2] The definition of throughput in bits is not given in this work.

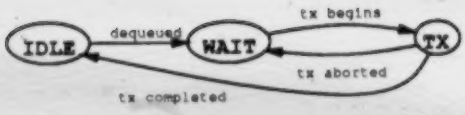


Figure 5: FSM representation of station behavior

3.5.2 Performance

Throughput

The interval between the moment the station leaves the state IDLE to the moment the station returns to the state IDLE is equal to the time taken to transmit one packet. Therefore, average throughput in packets per unit of time is $1/E(\Psi(tx\ completed))$.

To measure the throughput in bits per unit of time a model showing what happens to the individual bits is required. Note that throughput in bits per second is different from channel utilization.

Acquisition Delay

The acquisition delay, $\Psi(dequeued, tx\ completed)$ is the time a station waits to begin a successful transmission. This is the same as the acquisition delay from the point of view of the packet.

3.6 The Link Point of View

3.6.1 The Model

Assume that the link is composed of an array of cells each of which is able to hold one of four values representing a zero bit, a one bit, an unreadable value, and silence. The number of cells in a link is equal to the length of the link measured in bits. We assume that this number is an integer.

Given a state of the link, it can always be deduced what the next possible states are. This can be done from the simple rules of packet propagation. That is, we know that packets propagate in one direction in unidirectional links, and in both directions in bidirectional

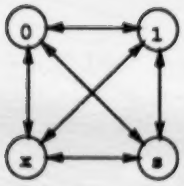


Figure 6: FSM representation of a single link cell.

links. We also know that packets are continuous and do not suddenly break up. This way we could construct a finite state machine partially describing the behavior of the link in order to define performance from this point of view.

3.6.2 Performance

Assuming that the propagation speed is constant, the intervals between state changes will be equal. Thus, measuring time intervals between the firing of transitions always results in a constant number. Therefore, performance from the link point of view cannot be defined as an interval performance measure.

Consider the behavior of a single cell making up the link. The behavior can be modeled with a four-state FSM as shown in Figure 6, where silence, zero, one, and unreadable bits are represented by the states **s**, 0, 1, and **x**, respectively. Several performance measures can be defined using the probabilities of finding the cell in a given state. The probabilities of finding the machine in state 0, 1, **s**, **x**, are $\Psi(0)$, $\Psi(1)$, $\Psi(s)$, and $\Psi(x)$. Of course, $\Psi(0) + \Psi(1) + \Psi(s) + \Psi(x) = 1$. The *utilization* of a single cell can now be defined as

$$u = \Psi(0) + \Psi(1) + \Psi(x) = 1 - \Psi(s)$$

It is not true that utilizations of all the cells are equal in general, however, the utilizations may be quite close especially in bidirectional links. Therefore we define the *average link utilization* as the expectancy of u taken over all cells in the link: $1 - \Psi_c(s)$. It is difficult to define just *link utilization*. In [53] link utilization is the fraction of total capacity used, but *total capacity* was not defined.

Observe that it seems impossible to define throughput as a performance measure from

the point of view of the link. We have defined throughput in packets per unit of time but throughput can be defined as the rate of useful bits transmitted. The link has no way of knowing whether the transmitted bit was useful or not, and so it seems that the more appropriate measure from this point of view is utilization.

3.6.3 Fairness

We assume that stations are connected to individual link cells rather than between cells. Therefore link utilization at a station is just the utilization of the cell to which the station is connected. We define a fair network from the point of view of the link as one in which the utilization of all cells is the same. Observe that if a network uses a unidirectional link then it must be unfair from the point of view of the link. On a unidirectional bus, channel utilization measured at downstream stations is always equal, or higher, than channel utilization measured at upstream stations.

3.7 The User Point of View

3.7.1 Performance

Performance from the point of view of the user can be either acceptable or not. For each traffic type we identify a performance measure which we believe is of greatest importance to the user, and then estimate the maximum acceptable value for this measure. Overall, performance is acceptable to the user according to the following proposition:

Proposition 2 (Performance from the User's Point of View) *Performance from the user's point of view is acceptable when the performance measures representative of each traffic type do not exceed a certain limit.*

Voice and Video

If voice packets are not late for playback the user has no way of telling how large the voice packet delay was. However, if the delay exceeds some threshold and some packets become

lost, the user will notice that the system is not working properly. A system to model the user's ability to detect lost packets can be constructed. This system must contain the transition *lost voice packet*. Observe that this system is not cyclic, since the user does not listen continuously. Whenever a packet is lost due to excessive delay, the transition *lost voice packet* fires. It is the frequency of the firing of this transition which is of interest. This frequency is sometimes expressed as ζ, the ratio of the number of packets received (not lost) to the number of packets transmitted (see Equation 2.1). In [17] it has been found that voice communication performance is acceptable to the user for values of $\zeta \leq 1\%$. Given that each packet holds 50 ms of speech, there are 20 packets/s transmitted while the voice signal is present. Thus, no more than 0.2 packets per second can be lost. This means that the minimum interval between lost packets is 5 s on the average. The interval between lost packets is $\Psi_c(\textit{lost voice packet})$. Therefore $\Psi_c(\textit{lost voice packet}) \leq 5$ s.

A similar model can be constructed for video traffic.

File Server

Whether the response time is acceptable or not will depend largely on the patience of the user, but it is always reasonable to require that the file server respond within some maximum interval of time. This model does not take into account the fact that the requests the user makes are often on a higher level. For example, the user may type the command **make** or cc, which will result in multiple requests to the file server. The user will probably be more interested in the performance of the whole command rather than a single request. Also, the user would expect a compilation to take longer for longer programs.

Even though transfer times are proportional to packet lengths, the acquisition time is usually the dominant delay. In addition, a long compilation may require many small requests to the file server, and the total time taken will be strongly influenced by the average response time of these requests. We propose the response time $\Psi(\textit{request sent}, \textit{reply received})$ as the file server performance measure from the point of view of the user.

Remote Procedure Calls

The obvious performance measure, from the user point of view, is the time delay from the moment a request is queued at the sender to the moment the reply is fully received at the same station. This can be written as $\Psi(request\ sent, reply\ recieved)$, where *request sent* and *reply received* are events in the finite machine representing user behavior.

Remote Logins

Similarly, in remote logins where characters are echoed, the user will likely require that characters be printed on the screen before the next key is struck.

This time is strongly influenced by the network delays because the processing time is very short. Therefore, we choose $\Psi(char\ sent, char\ echoed)$ as the performance measure most important to the user during remote logins.

3.7.2 Fairness

We have defined fairness as the ratio of maximum to minimum performance. This definition is useful for FS, RPC, and remote login traffic, but in cases where we are only interested whether performance is acceptable at all stations, another definition may be more appropriate. That is, if performance is acceptable at all stations then the network is fair.

Definition 7 (Fairness (alternative definition)) *Fairness is the ratio of the number of stations where performance is not acceptable to the number of stations where performance is acceptable.*

CHAPTER 4

Simulation Studies in Behavior of Unfair Protocols

4.1 Introduction and Summary of Results

All simulation results presented in this chapter have been obtained using a simulator called LANSF [16]. The simulator uses a very accurate model of a local area network and few simplifications are made. For example, packets propagate in the same way as on a real link. Stations can detect collisions, silence, beginning and end of transmission, and other events. In LANSF, a protocol is a collection of processes programmed to respond to such events. The different traffic types were implemented by creating special processes to generate packets, requests and replies according to the parameters given in Chapter 2.

The first part of this chapter studies the effect of traffic type and a (the ratio of bus length to packet length) on fairness. Fairness from the packet point of view is measured under uniform one–way traffic, and then compared under file server, remote procedure call, and voice traffic. It is found that the acquisition delay curve changes with traffic type. Also, the fairness under traffic types in the request/reply category is found to be influenced by a. Namely, increasing a improves fairness. This is especially apparent for file server traffic. This result is very important. It shows that unfair protocols are well suited for this kind of traffic on fast networks.

In the second part of the chapter FS traffic is studied. Emphasis is placed on comparisons between different variants of LCSMA–CD. Slotted LCSMA–CD is compared with the unslotted version. It is apparent that the slotted version should not be as unfair as

the unslotted version. Given an empty slot, a station is guaranteed to transmit its packet successfully, regardless of the position of the station. In the unslotted case, the station can always be preempted. The difference will be especially apparent in the case of long packets containing files. The disadvantage of slotting is that (short) request packets only partially fill the slot, and there are many request packets. The advantage of slotting can be easily carried over to the unslotted version by limiting the maximum packet length. The effect of maximum packet length and slot size on performance and fairness is examined. The effect of giving preference at the file server to stations according to their position on the bus is also investigated. For example, if the server is placed in position 0 (the most privileged position on the network) and more than one request has arrived to the server, the request from the station farthest removed form the server would be serviced first, regardless of the order of arrival of the requests. Another modification to the protocol executed by the file server is also studied, given that the server is placed in the most privileged position. The file server is allowed to transmit only when it senses silence on its input port. Since the servers are the stations that transmit the majority of the packets on the network, this modification improves the fairness of the protocol.

The effect of positioning the file server in the least and most privileged positions is also studied. Placing the server in the least privileged position has a positive impact on fairness. This is because the server cannot preempt transmissions from any other station, and it effectively uses the idle cycles of the network. That is, the server transmits only when no other station has a packet to transmit.

The effect of service time on fairness is also studied. Service time is defined as the time the file server takes to service a request on the average. Longer service times improve fairness. Since the unfairness of the network is due to uneven acquisition times, and response time does not have a pronounced effect on the acquisition time, increasing the service time will increase the queuing time and service time more or less evenly for all stations.

The file server model is used as a test environment for comparing the performance of LCSMA-CD with two representatives of the DAMA family: Expressnet for the single bus, and FASNET for the double bus. The superiority of LCSMA-CD is clear; moreover, its

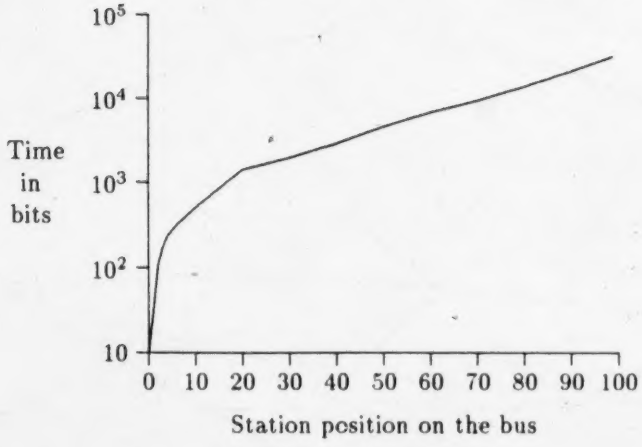


Figure 7: Performance from the packet point of view (acquisition time) under uniformly distributed one–way traffic for LCSMA–CD on the single unidirectional bus.

supposedly obvious unfairness disappears almost completely for this traffic model.

4.2 Fairness versus Traffic Type

4.2.1 Uniform Traffic

Uniform traffic is the most commonly studied traffic type. This is not because it is common for a network to carry this kind of traffic, but rather because it is well suited to analytical modeling. Figure 7 shows performance from the packet point of view for LCSMA–CD on the single unidirectional bus. The acquisition time increases as the station position index increases. The simulation parameters are as follows. The mean message length is set to 8192 bits, and the traffic intensity is adjusted to near maximum. The preamble length is 16 bits, and the header length is 208 bits. The maximum packet length is limited to 4096 bits. The most privileged station is station zero and the acquisition time at that station is also zero. The acquisition time increases with the position of the station on the bus. Simulations performed showed no relationship between fairness and a.

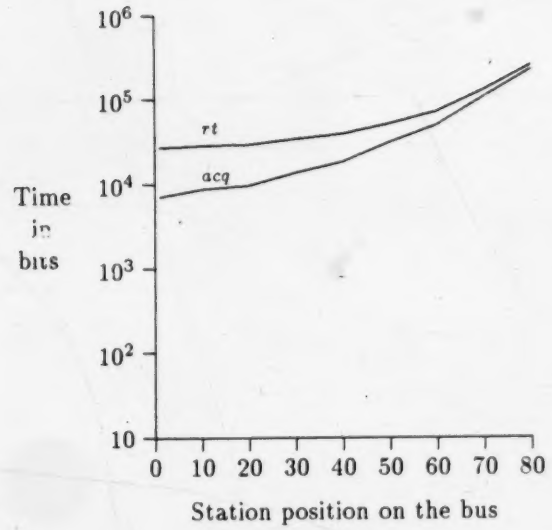


Figure 8: Performance from the packet point of view (acquisition time *acq*), and the user point of view (response time *rt*) at different stations for single bus unslotted LCSMA-CD under FS traffic with the server in the most privileged position.

4.2.2 File Server Traffic

The simulation parameters are as follows. Requests are generated 10 times per second. This is faster than what would normally be the case in a real network where about 4 requests per second would be made. However, if client stations make only 4 requests per second then a large number of stations and long simulation times may be required. Other parameters in Figures 8 and 9 are intended to approximate a 10 Mbit/s network. The read request and acknowledgement packets are 32 bits long. The maximum packet length is 4092 bits, and the mean file length is 8192 bits and the interstation gap is 3 bits.

The results for file server traffic are quite different from those for uniform traffic. It is not only that the position of the file server affects fairness. The shape of the acquisition time curve for FS traffic with the FS in the most privileged position is different than that under uniform traffic. The most striking result is that the network is fair from the user point of view when the file server is in the least privileged position.

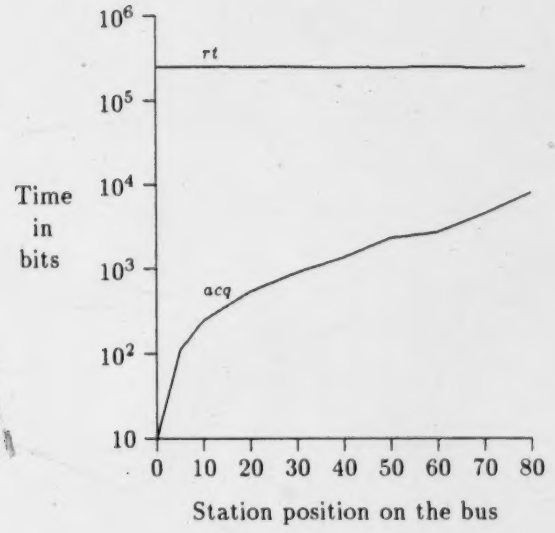


Figure 9: Performance from the packet point of view (acquisition time *acq*) and user point of view (response time *rt*) for LCSMA-CD on the single bus under FS traffic with server in the least privileged position.

The latter can be explained as follows. Since the file server must yield to all other transmissions, which are always requests, the size of the request queue at the file server will normally be larger than one. Of course, when the traffic intensity is low enough, the queue length will be zero, but then any network is fair at very low traffic intensity. If the queue size is larger than one then a significant portion of the response time will be attributable to the time the request spends in the request queue. This will be amplified by the length of the acquisition time which is largest at the file server station. In addition, since the file server uses about 80% of the bandwidth, and packets transmitted from the file server do not preempt any other packets, a station which is transmitting a request stands little chance that it will have to abort its transmission due to a packet from upstream. Because preemptions are less likely, the acquisition times measured at individual stations will be lower and therefore will differ by a smaller absolute amount. Also, the service at the file server is more or less round robin. To see this, consider a network with only three client stations: s_0, s_1, and s_2, where s_0 is the most privileged station. Suppose also that the traffic

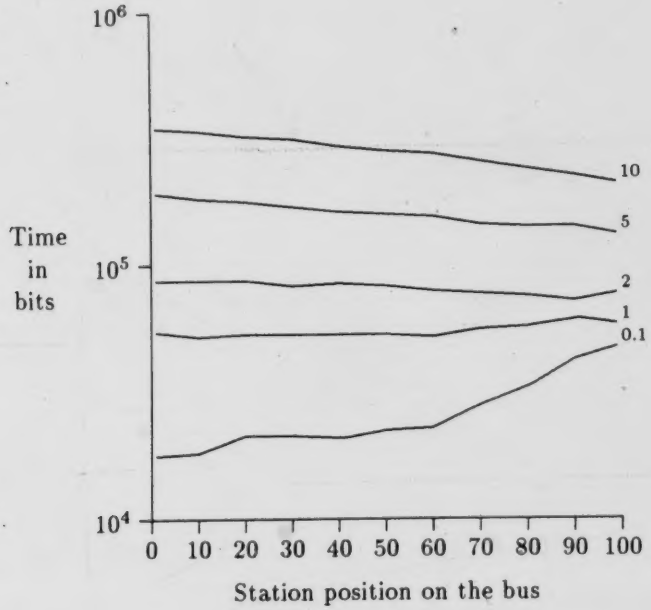


Figure 10: Response time for LCSMA–CD on the single unidirectional bus with FS in most privileged position for different values of a.

is very heavy. That is, the client stations do not wait before transmitting the next request, and the service time at the file server is also zero. The client stations will transmit in the order of proximity to the file server, but all will be able to transmit before the first reply comes. As soon as s_0 receives its reply it resubmits its request. Then the same happens with s_1, and s_2. Therefore, the requests are effectively serviced on a round robin basis at the file server. When the traffic is not so heavy, the service discipline is not round robin but approximates it to a certain degree. This is because any backlogged station is guaranteed to submit its request before any other station receives a reply. Of course, this argument holds only for small values of a.

The fairness with this traffic type depends on the ratio of packet length to the bus length. Suppose that stations are numbered from the left to right from zero to $n-1$, where n is the number of stations on the single unidirectional bus. Station zero is the most privileged station. Observe that packets transmitted from this station travel the greatest distance on

the average if all stations are equally likely to be chosen as receivers. This is also the case if the station communicates only with the file server. Suppose further that the distance from station 0 to station $n-1$ is d, and that stations are spaced equally with the distance between two neighboring stations being Δd. The collective propagation delay component of a request and a reply packet for stations with higher indices is smaller than that for stations with low indices. In particular, station $n-2$ has the smallest propagation delay component and therefore the response time at that station is less sensitive to a than at other stations. Observe that the distance a request must travel (propagation time for a request packet) is the same as the distance the reply must travel. If the file server is in the most privileged position, then the propagation delay component for station i is $2(d+(n-1-i)\Delta d)$, and $2(n-1-i)\Delta d$ if the server is in the least privileged position. Figure 11 shows clearly that when the FS is in the least privileged position the response time increases for stations with low indices faster. It is difficult to argue that this unfairness is undesirable. It is very natural to suppose that stations physically placed farther from the file server should wait proportionately longer. When the file server is in the most privileged position the fairness from the user point of view improves with increasing a as shown in Figure 10, but the network does not become unfair to stations with low index values as quickly. For the dual bus, the distance that a request and the the reply must travel for station i is $2|\frac{n}{2}-i|\Delta d$ if the file server is in the least privileged position, and $2i\Delta d$ if the file server is in position zero. Since the station with smallest acquisition time will experience the highest propagation delays, similar results will be obtained for the dual bus.

Two more observations are in order. First, the average link utilization decreases[1] with increasing a because the propagation delays become dominant. And second, it is possible to make the protocol fair, regardless of the load, by adjusting a appropriately because fairness improves with increasing a regardless of the load. Since higher values of a imply higher bandwidth (or a longer bus) we conclude that fairness will be less significant in very high bandwidth networks carrying this traffic type.

[1] Assuming that the number of client and server stations is kept constant.

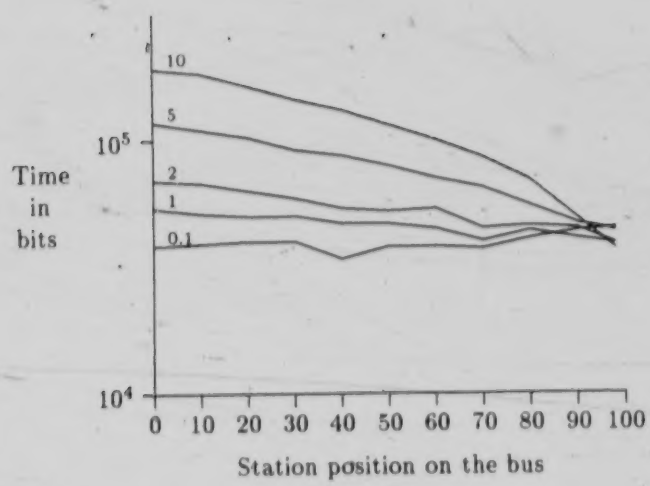


Figure 11: Performance from user point of view (response time) for LCSMA-CD on the single unidirectional bus with FS in least privileged position for different values of a.

4.2.3 Voice Traffic

The simulation parameters are as follows. Packet speech content is set to 50ms, and the digitizing rate to 64 kbits/s. This means that packets are 3200 bits long. The parameters here are intended to approximate a 100 Mbit/s network.

The network is fair for this kind of traffic from the user point of view as long as packet delays do not exceed a certain maximum for more than 1% of the time. Figure 13 shows the acquisition time cumulative distribution function for the least privileged station on the bus. It is assumed that the acquisition times at any other station on the bus are worse than that. Figures 13 and 12 show curves for 1000 and 1500 conversations. Callers and called stations were chosen at random with uniform distribution. No packets were lost in either case since d_0 was chosen to be 40 ms ($4 \cdot 10^6$ bits). In this case, as in the case of FS traffic with the file server in the least privileged position, the network is fair from the user point of view but not fair from the packet point of view.

Simulations showed no relationship between fairness and a for this traffic.

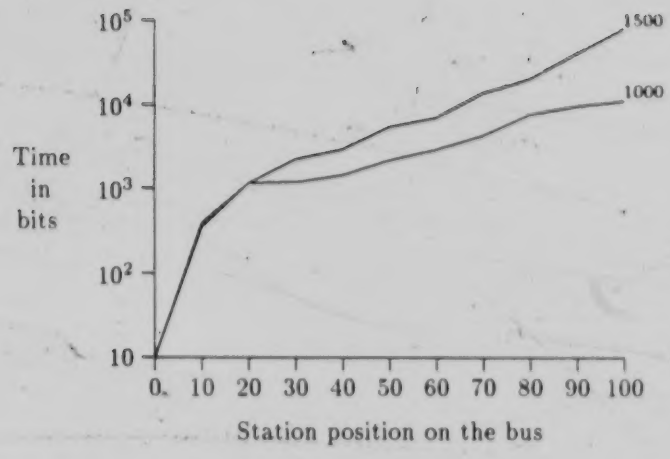


Figure 12: Performance from the packet point of view (acquisition time) for LCSMA-CD on the single 100 Mbit/s unidirectional bus carrying 1000, and 1500 conversations.

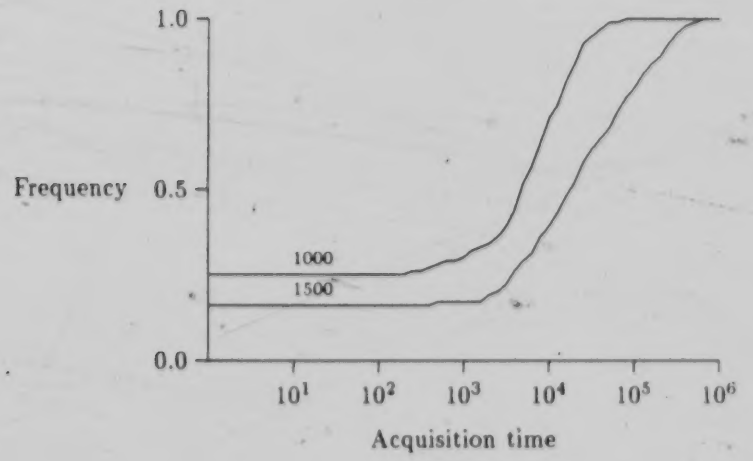


Figure 13: Acquisition time cumulative distribution function at the least privileged station on a 3000 bit long cable with 100 stations and 1000, and 1500 conversations.

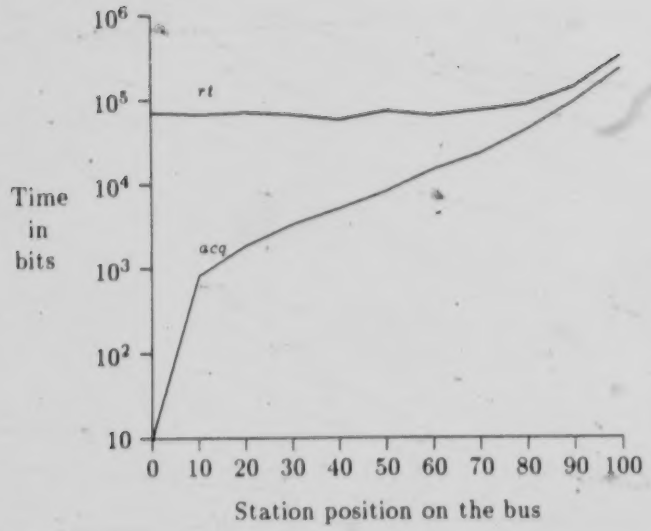


Figure 14: Performance from the packet point of view (acquisition time *acq*) and user point of view (response time *rt*) for LCSMA-CD on the single unidirectional bus under RPC traffic.

4.2.4 RPC Traffic

The simulation parameters were taken as follows. Service time was set to zero, and request generation time was set such that the total throughput would be around 0.5. Mean length of the request packet was set to 256 bits, and mean length of the reply to 8192 bits, both exponentially distributed. In Figure 14, $a \approx 0.1$.

The network under this traffic type is again unfair from the user and packet points of view. From Figure 14, it is clear however that user unfairness is nowhere near as severe as packet unfairness.

On the single unidirectional bus, the average propagation delay for station i is $2((n - 1 - i)\Delta d + \frac{d}{2})$. For the dual bus the average delay is $|\frac{n}{2} - i|\Delta d$. Since the more advantageous the position of the station the higher the propagation delay, the fairness improves with increasing a. Figure 15 shows this clearly.

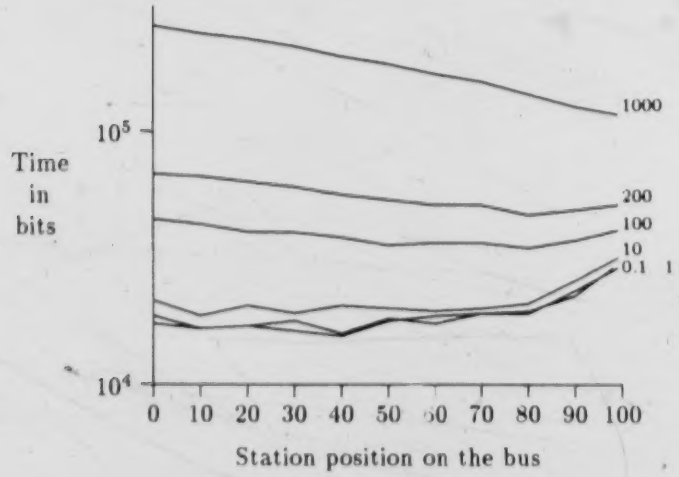


Figure 15: Response time for LCSMA-CD on the single unidirectional bus for different values of a under RPC traffic.

4.3 Behavior of Unfair Protocols Under File Server Traffic

4.3.1 Single Unidirectional Bus

Simulation Parameters

As in Section 4.2.2, the read to create ratio is 4:1, the mean file length is 1K bytes, and the frequency of requests is 10 request per second (assuming zero response time).

The network load is changed by varying the number of stations connected. Each station has a certain bandwidth requirement. This requirement, however, is not constant because the frequency of packet transmissions is dependent on the delays, and therefore, the traffic. In the simulation experiments conducted in this section, networks under loads of 40, 60, 80, stations on the single unidirectional bus are studied, and also 100 stations on the dual unidirectional bus. This corresponds to total throughput of about 0.32, 0.48, 0.64, 0.8, respectively. Results for 80 and 100 stations are only reported. That is, measurements shown in graphs were taken under high load conditions[2].

[2]In all cases the total throughput was above 0.6.

The distance between stations is kept constant. Networks with two different interstation gaps are studied. The first one is set to 3 bits. This kind of network, together with all other parameters as above, is intended to represent a 10 Mbit/s network. Ideally, to convert the model to a 100 Mbit/s network, one should divide the bandwidth requirement of each station by 10, increase the number of stations 10 times, and increase the interstation gap 10 times. Unfortunately, such a modification results in a very large number of stations and very lengthy simulation times. Therefore, we leave the number of stations the same and only increase the distance between stations. Thus, for example, a 10 Mbit/s link with 80 stations is 240 bits long, whereas a 100 Mbit/s link is 2400 bits long.

LCSMA–CD

File Server in Least Privileged Position When the file server is put in the least privileged position it is allowed to transmit a packet only when no other station has anything to transmit. Thus, it can be thought of as using the idle cycles of the network. Given a read to create ratio of 4:1, the file server uses almost 80% of the bandwidth. Such a scenario can be very useful in a situation where data traffic is to be given lower priority over some other kind of traffic such as voice.

File Server in Most Privileged Position When the file server is in the most privileged position it is able to transmit the reply as soon as it is finished servicing a request.

Suppose that the service time is zero. The response time is equal to the acquisition time plus the transmission and propagation delays. This network is obviously unfair because the stations closer to the file server have a shorter acquisition time. The file server queue rarely exceeds the length of 1 because the reply is made as soon as the request arrives. Unfortunately, this configuration can lead to starvation. When the operating point approaches saturation, stations in the least privileged positions may not be able to transmit packets.

Increasing the service time is expected to make the network more fair because of the increased queue length and the increased amount of time requests spend in the queue at the file server. Making the service time very long results in a network which is completely fair

because all stations have requests waiting in the queue and the file server treats all requests with equal priority.

Variants of LCSMA–CD

LCSMA–CDQ The unfairness of LCSMA–CD with the file server in the most privileged position can be compensated for by exchanging the order of requests in the file server queue. The rationale behind this is that stations most remotely positioned from the server have waited long enough to acquire the bus, and therefore should not be required to wait in the service queue. Thus, every time a new request arrives, the queue is sorted such that requests from stations farther away from the server will be serviced sooner than stations closer to the server. This modification in no way improves the performance of the network, but it might improve the fairness if the service time at the file server is not unreasonably long. If the service time is excessive, then the network will discriminate against the stations close to the server.

LCSMA–CD2 Once again consider the case where the file server is put in the most privileged position. When the file server transmits a reply, it will sometimes damage one or more of the incoming requests. If the length of a request packet is long compared with the length of the link, this scenario will be more likely than in the case where packets are short. The damaging effect of file server replies can be rectified by requiring the file server to wait until there is silence on its input port. The savings should be especially apparent under heavy traffic conditions where several requests strung together are more likely to arrive. Also, observe that the create request packets can be quite long. Modifying the protocol this way should result in a substantial reduction in the number of destroyed create requests. The modification should improve the fairness of the protocol because there will be less destroyed packets and less noise on the link, giving the less privileged stations a better chance for transmitting their packets. In addition, the performance should improve. Of course, for very fast networks, there will be little or no difference between LCSMA–CD and LCSMA–CD2.

maximum	*cable length*	
packet length	240	2400
1024	0.13	0.35
2048	0.19	0.39
4096	0.20	0.41
8192	0.23	0.23
16384	0.17	0.19
32768	0.13	0.12
unlimited	0.09	0.11

Table 4.5: Effect of packet length on fairness for the 240 and 2400 bit cable, with 80 stations and no service time.

LCSMA–CDQ2 The difference between LCSMA–CDQ2 and LCSMA–CD2 is the same as the difference between LCSMA–CD and LCSMA–CDQ. This modification is expected to improve the fairness because LCSMA–CD2, even though not quite as unfair as LCSMA–CD, is nevertheless an unfair network.

LCSMA–CD/S The performance of the protocol may change when slotting is introduced. The advantage of slotting is that collisions are eliminated. Collisions unnecessarily consume bandwidth which otherwise could be used to carry legitimate packets. On the other hand, the disadvantages of slotting are that small packets do not fit in their entirety into the slots, and that large packets are transmitted in several smaller packets requiring more headers.

Effect of Packet Length

LCSMA–CD with FS in Most Privileged Position Since very long packets stand a good chance of colliding, shortening the maximum packet length reduces the chances of collision and therefore improves the fairness of the protocol. However, very short packets have a large amount of overhead due to the header length. For read request packets and acknowledgements, this overhead amounts to about 50%, whereas for create requests and read replies it amounts to about 2%. Figure 16 shows that best results are achieved when the maximum packet length is around 4000 bits on the 2400 bit long network. For the 240 bit long network best results were obtained with the same maximum.

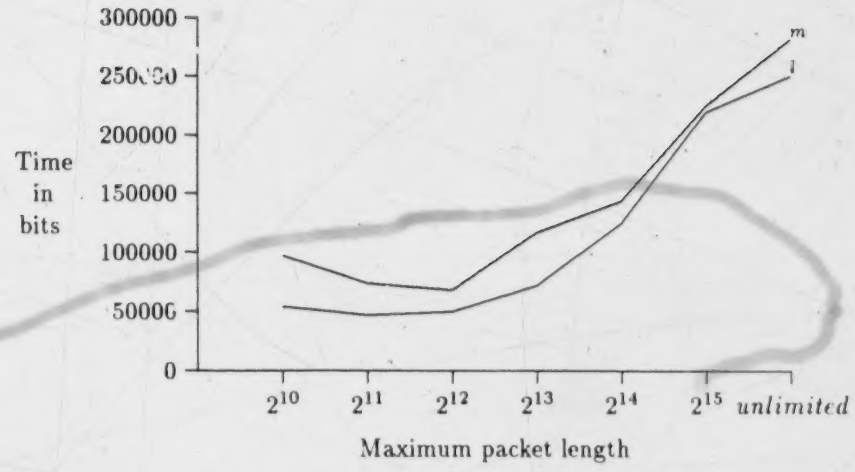


Figure 16: Effect of maximum packet length on response time in LCSMA/CD, with file server in the most privileged position m, and the file server in the least privileged position l, 80 stations, no service time, 2400 bit cable.

Increasing the cable length improves the fairness of the protocol. The reason for this is that longer cable length implies shorter relative packet length without the added overhead due to headers. Therefore, packets stand less of a chance for collisions on a longer cable. Also, on a longer cable, packets from more privileged stations have to travel farther. The results are shown in Table 4.5[3].

LCSMA–CD with FS in Least Privileged Position Most long packets are transmitted from the file server. The acquisition time, and therefore the service time, is decreased when the maximum packet length is limited. Measurements indicate that if the packet length is not limited then the response times are about 250,000 bits, with 80 stations, no service time and a 240 bit cable. Again, very short packets give worse performance than packets of moderate size due to the header overhead. Shortest delays are achieved with maximum packet length of 2000 to 4000 bits. On the 2400 bit cable, minimum delays are achieved when the packet length is limited to about 2000 to 4000 bits also.

[3] Fairness was calculated as the ratio of the minimum to the maximum response time measured.

servers	*Expressnet*	*LCSMA-CD*
0	117000	51000
2	65000	50000
4	50000	51000
8	48000	50000
16	49000	52000

Table 4.6: Average response times for Expressnet and LCSMA-CD with file servers in the least privileged positions. The interstation gap is 3 bits, and the maximum packet length in LCSMA-CD is 4096 bits. There are 80 client stations in each case.

LCSMA-CD/S Long slots in slotted LCSMA-CD result in a waste of bandwidth due to the fact that many packets are not long enough to fill the slot. Short slots carry a larger proportion of header information. Simulation results show that shortest delays are achieved with a slot length of 1024 for LCSMA-CD/S with the file server in most and least privileged positions.

Comparative Performance

LCSMA-CD with FS in the Most Privileged Position For low traffic intensities, the performance of this protocol is comparable to Expressnet and other variants of LCSMA-CD. For higher traffic intensities, the performance of the stations farthest removed from the file server deteriorates substantially (see Fig. 17).

Since the performance of the protocol is less dependent on a than the performance of Expressnet, the results in Figure 18 are not surprising. Protocols which use round robin scheduling, will eventually be surpassed by this variant of LCSMA-CD given a sufficiently large transmission rate.

LCSMA-CD with FS in the Least Privileged Position This protocol performs better than its slotted version LCSMA-CD/S with the file server in the same position. For the 10 Mbit/s network, however, the performance of this protocol is inferior to that of Expressnet. For the 2400 bit cable, the performance of this protocol is substantially better than that of Expressnet.

Placing the file server in the least privileged position results in longer delays for most sta-

tions, however, the most distant stations from the server have delays which are significantly lower.

LCSMA–CDQ Switching the order of service at the file server depending on the position of the station appears to have a positive impact only on performance at stations moderately removed form the file server. Stations very close to the server will have delays larger than those recorded with LCSMA–CD. To see this, consider the station next to the server as an example. Whenever a request from this station is found in the server queue with another request, this request will be serviced last. Under high traffic conditions, stations farthest away from the server benefit rather little from the high priority they receive at the server. The reason for this is that the dominant component of the response time for these stations is the acquisition time rather than the delay time at the server. Under low traffic conditions, or conditions where queues at the server are short or nonexistent, the performance of this protocol approximates that of LCSMA–CD.

LCSMA–CD2 and LCSMA–CDQ2 The performance of LCSMA–CD2 is not quite as good as that of Expressnet for slow networks. However, when the value of a decreases, the response times in LCSMA–CD2 are lower for all stations (see fig. 18.) This, again, can be attributed to the fact that Expressnet uses trains to transmit information and is therefore sensitive to a. LCSMA–CD2 is clearly better than Expressnet for transmission speeds of 100 Mbit/s and higher. LCSMA–CD2 also compares favorably with LCSMA–CD and LCSMA–CD/S. Stations in LCSMA–CD2, which are close to the server, have delays larger than the corresponding stations in LCSMA–CD. However, LCSMA–CD2 is not nearly as unfair as LCSMA–CD. The fairness of LCSMA–CD2, with 80 stations on the 10 Mbit/s network, with no server delay is 0.69, whereas the fairness of LCSMA–CD under the same conditions is 0.23. For higher speed networks, the performance of LCSMA–CD and LCSMA–CD2 should be quite similar, regardless of service time. If packets are short, the extra amount of time that the file server will have to wait will be negligible.

As expected, the performance of LCSMA–CD2 and LCSMA–CDQ2 is very much the

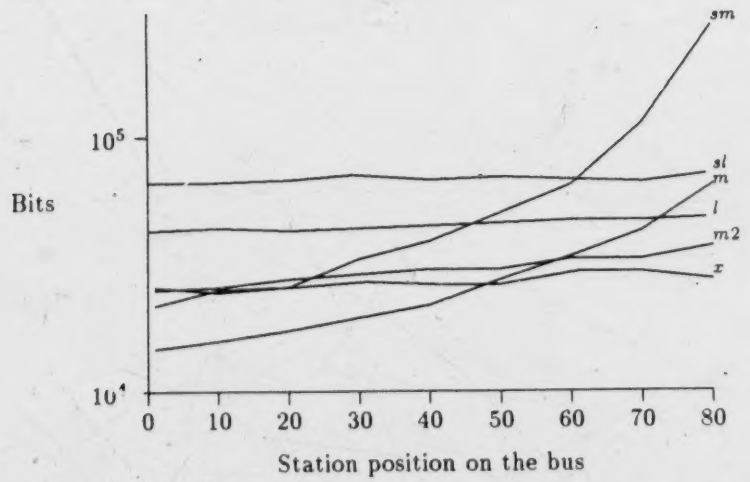


Figure 17: Response time with 80 stations, no service time, on the 240 bit cable, for Expressnet (x), LCSMA-CD/S with server in least privileged position (sl), LCSMA-CD/S with server in the most privileged position (sm), unslotted LCSMA-CD with servers in most (m), and least (l), privileged positions, and LCSMA-CD2 ($m2$).

same for slow networks and zero service time. The performance of these two protocols differs for slow to moderately fast networks only if the service time is sufficiently long. The difference lies in which stations are the privileged ones rather than in the overall performance. However, if the service time becomes excessively long, the fairness of LCSMA-CDQ2 is much worse than that of LCSMA-CD2.

LCSMA-CD/S This variant of LCSMA-CD performs worse than the original. With the packet length set to 1024 bits, for every request or acknowledgement packet there will be about 8 packets transmitted containing the portions of the file. Since the last packet may also be only partially filled, there should be about 2 partially filled packets for every 8 completely filled packets. The problem must therefore lie with the amount of header information transmitted. Other simulation runs showed that increasing the slot length does not appear to improve the situation at all. If the slot length is set to about 2000 bits, then there will be about 2 partially filled packets for every 4 completely filled packets. Simulation

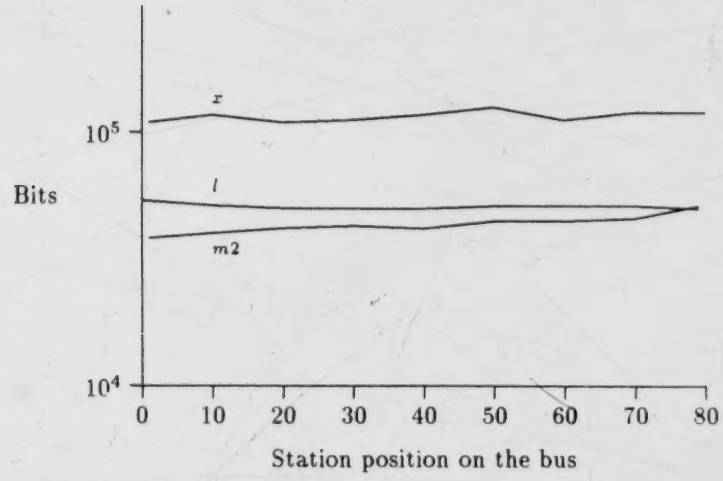


Figure 18: Response time with 80 stations, no service time, on a 2400 bit cable for Expressnet (x), LCSMA-CD with server in least privileged position (l), and LCSMA-CD2 ($m2$).

results indicate that increasing the packet length in LCSMA-CD/S has a detrimental effect on its performance.

Expressnet and the Number of File Servers The reason for the poor performance of Expressnet on the 2400 cable is that the file server is permitted to transmit only one packet per round. This increases the service time at the file server, and therefore the average queue lengths are longer. The situation can be remedied by adding more file servers. The results are shown in Table 4.6.

If the cable length is increased even further, performance of Expressnet becomes inferior to the other protocols even if the number of servers is large. Measurements with 80 stations, 16 file servers, and an interstation gap of 50 bits (cable length of 4750 bits), determined the average response time for Expressnet to be 150,000 bits. For LCSMA-CD with file servers in the 16 most privileged positions, the average response time at station 16 was 85000 bits, and the delay at station 95 was 105,000 bits, using packets limited to 8192 bits.

Sensitivity to Service Time

LCSMA–CD with File Server in most Privileged Position The unfairness of LCSMA–CD, with the file server in the most privileged position, is affected by the service time of the file server. For low to medium traffic intensities, the fairness improves with increasing service time. When the service time is zero, every read request is followed by a file transmitted from the server, assuming that the length of the read request is larger than the length of the link. When a file is transmitted from a station to the server the situation is very much the same. Stations closer to the file server have a lower response time because they have a lower acquisition time.

When the service time increases, the file server may receive more than one request before it is able to transmit a reply. However, the frequency of arrival of the requests must be equal to the frequency of replies. In other words, if the mean service time is larger than the mean request arrival time, the length of the request queue at the file server will equal to the number of stations on the network. If this were the case then the file server would be the bottle–neck, and in real life situations another file server would have been added.

Suppose that the service time is such that n request could be transmitted on the average before the file server makes a reply. If, at a given moment in time, there are $m < n$ backlogged stations, then all of them will be able to transmit their requests. Even though the stations further away from the server will still be serviced last, their acquisition time will be reduced because now they cannot be preempted by stations closer the file server which become backlogged. For low to medium traffic intensities $m < n$ will be true almost always, and the network becomes more fair. In contrast, for high traffic intensities, it may commonly occur that $m > n$. The stations furthest removed from the file server will not benefit as often from the increased service time as stations closer to the file server. It is not clear that the network should become more fair. Figure 19 shows the effect of increasing service time on fairness in case where the maximum packet length is limited to 4092 bits, and where there is no limit on the packet length. It is apparent that the fairness improves as the service time increases when the packets are limited, but in the other case this is not

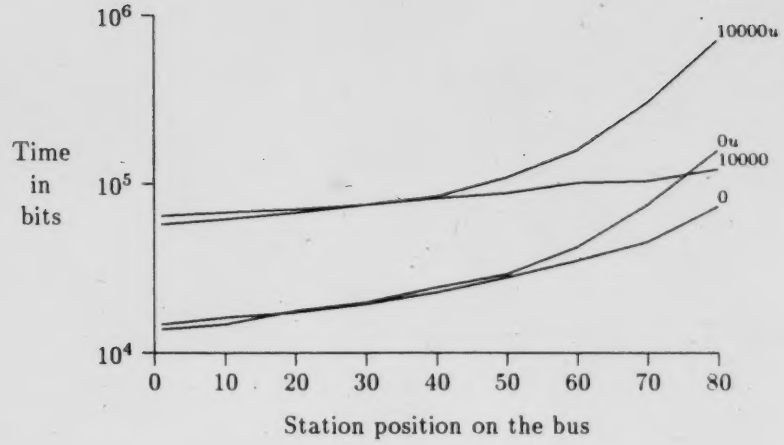


Figure 19: Effect of service time on response time in LCSMA–CD, with maximum packet length of 4092 bits, and unlimited length u, on a 240 bit cable, with service times of 0 and 10000 bits.

so clear.

LCSMA–CD with File Server in least Privileged Position Increasing the service time under low traffic intensities has little effect, other than increasing the response time by amount slightly greater than the service time. Under high intensity traffic, increasing the service time makes no difference because the file server acquisition time is very long, and a request can be serviced while a reply awaits transmission. The increase in service time becomes apparent only when the service time exceeds the acquisition time often enough. When this happens, the effect on the response time is multiplied due to the fact that the average queue length is not zero.

LCSMA–CDQ and LCSMA–CDQ2 It is obvious that these protocols will favor the stations furthest removed from the file server when the service time is such that the server becomes a bottleneck. Surprisingly, simulation results show (Figure 20), that the protocols can become unfair to the stations near the server even if the request queue length is moderate. For example, for heavy traffic intensity (80 stations), and service time of 1 ms

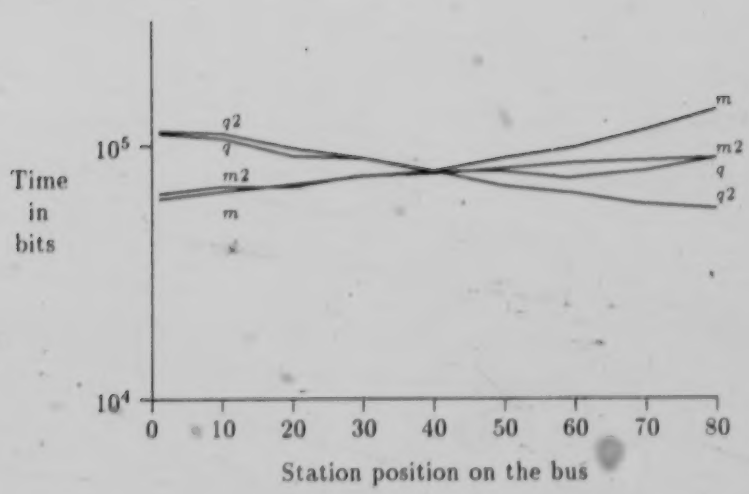


Figure 20: Response time with 80 stations, 1 ms service time, on a 240 bit cable for LCSMA-CDQ q, LCSMA-CDQ2 $q2$, LCSMA-CD m, and LCSMA-CD2 $m2$, with the file server in the most privileged position.

(10000 bits), the fairness of LCSMA-CD with file server in the most privileged position is 0.52 favoring the stations close to the server, whereas the fairness of LCSMA-CDQ is 0.80 favoring the stations far away from the server.

Overall, the performance of LCSMA-CDQ is better than that of LCSMA-CD, but LCSMA-CD2 is better than LCSMA-CDQ2 for service time of 10000 bits, on the 240 bit cable.

4.3.2 Dual Unidirectional Bus

LCSMA-CD

File Server in the Most Privileged Position Suppose that the file server is the left-most station. Then, all the requests will be transferred on the right-to-left (RL) link, whereas all replies from the server will be transferred on the left-to-right (LR) link. Since there are approximately four times as many bits transmitted by the server than by the client stations, under very heavy traffic, the LR link will become a bottleneck. Increasing

the traffic intensity further will result in long queues forming at the file server, making the network more fair. This line of reasoning leads us to the conclusion that the unfairness of this protocol is bounded under these assumptions. In addition, this bound is dependent on the read to create request ratio. The fewer create requests there are, the fairer the network.

Observe also that the maximum throughput on the RL link can only be about 0.25 at the maximum, given a read to create ratio of 4:1. It is rare for local area networks to operate at maximum load for prolonged periods of time. This implies that unfairness of the network cannot be severe because unfairness is not apparent under low traffic intensities.

File Server in the Least Privileged Position Suppose that there are N stations on the bus and that they are numbered left to right from 1 to N, and that the file server is in the middle. Since packets are transmitted between the server and the clients and never between the clients, stations $\{1, \ldots, N/2 - 1\}$ do not transmit on the RL link, and likewise stations $\{N/2 + 1, \ldots, N\}$ do not transmit on the LR link. For this reason, transmissions from the file server cannot interfere with requests from client stations. The resulting network is not fair. Arguments given in section 4.3.1 to support the conclusion that LCSMA-CD (on the single bus) with the file server in the least privileged position is fair do not hold because the file server transmits on a different bus than the client station making a request. In fact, simulations show that response times for stations near the middle of the bus are slightly higher than for stations near the ends of the bus. The unfairness of the network is so small that it is hardly noticeable. For our purposes, it is reasonable to assume that this network is fair. Incidentally, FASNET is not a strictly fair network either. Simulations show that if the file server is placed in the middle, the response times for stations in the middle are slightly higher than the turnaround times for stations near the ends.

Effect of Packet Length As in the case of the single unidirectional bus, shortening the packet length reduces the chances of collision for long packets. The effects can indeed be quite dramatic. Measurements indicate that packet lengths of about 4000 bits give best response times for LCSMA-CD with the file server in the most privileged position.

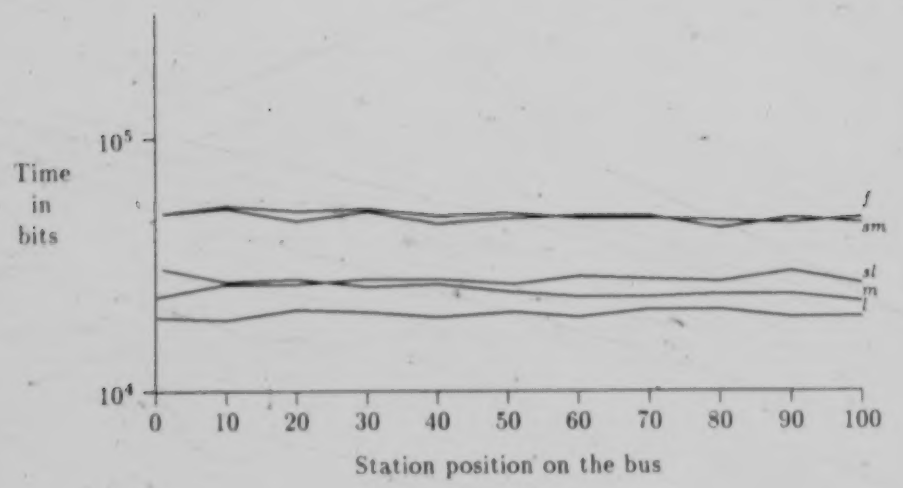


Figure 21: Response time with 100 stations, no service time, on a 300 bit long cable. For FASNET f, LCMSA-CD with server in least l and most m privileged position, and LCSMA-CD/S sl and sm.

Comparative Performance

FASNET

The FASNET protocol used here is slightly different than the original. In the original, a station is allowed to transmit only one packet in a given round. This leads to very long delays at the file server. Therefore, we have modified the protocol to allow any station to transmit all of its pending packets in a given round.

Figure 21 compares LCSMA-CD, slotted LCSMA-CD, and FASNET. The file server in FASNET is placed in the most privileged position and the packet length is set to 1024 bits. Increasing the packet length and moving the file server to the least privileged position did not improve the performance of FASNET.

Adding more file servers improves the performance of FASNET because the service time at the file servers is reduced. On the 300 bit cable, however, the rounds are relatively short. The average response time with one file server was measured to be 57000 bits, whereas the delay with 16 file servers was measured to be 55000 bits with other parameters as in

Figure 21 and an interstation gap of 3 bits.

LCSMA-CD/S

The slot size in LCSMA-CD/S is the same as that in FASNET. Increasing the slot size appears to have a negative impact on the response time. LCSMA-CD/S with the file server in the most privileged position is essentially the same as FASNET with the file server in the most privileged position. The difference between the two protocols is that the file server in FASNET must wait for about 1000 bits before transmitting the next reply. As expected, the performance of the two protocols is very similar on the 10 Mbit/s network.

Moving the file server to the least privileged position improves the performance because the file server is sometimes able to transmit on both links at the same time. Simulation results show that queues at the file server are shorter when the file server is placed in the middle.

LCSMA-CD

The unslotted version of the protocol performs better than both FASNET and slotted LCSMA-CD. This is tied to lower service times at the file server. For example, if the file server is placed at most privileged position, then the service time at the file server in LCSMA-CD/S will be about 8 times the header length.

CHAPTER 5

Summary and Conclusions

5.1 Contents Summary

5.1.1 Chapter 2

Chapter 1 is a survey of traffic types commonly carried by broadcast bus networks. The traffic types surveyed are:

- File Server Traffic
- Remote Procedure Call Traffic
- Voice Traffic
- Remote Login Traffic
- Video Traffic

A simulation model (with parameters) is given for FS, RPC, and voice traffic.

5.1.2 Chapter 3

In Chapter 3 we formally define fairness on the assumption that if a network is fair then there are differences in *performance* measured at different stations. Since fairness is closely related to performance we introduce the following classification (and definition) of performance measures.

- *Performance Measures from the* packet *point of view.*

- Queuing Delay
- Acquisition Delay
- Propagation Delay

- *Performance measures from the* station *point of view.*
  - Throughput
  - Acquisition Delay

- *Performance measures from the* link *point of view.*
  - Channel Utilization

- *Performance measures from the* user *point of view.*
  - Response Time
  - Interval Between Consecutive Packet Losses

The following observations were made. Fairness can be measured in many different ways. Unfairness from one point of view does not necessarily imply unfairness from another. In addition, in Section 3.6.3 it is observed that any unidirectional bus must be unfair from the link point of view.

5.1.3 Chapter 4

In Chapter 4, a large number of simulation experiments is performed. The following problems are investigated:

- Dependence of fairness on traffic type.
- Dependence of fairness on point of view.
- Dependence of fairness on a.
- Behavior of unfair protocols under file server traffic.

The following observations were made:

- *Fairness from the same point of view depends on the traffic type.* In Section 4.2 the acquisition time curve for all stations on LCSMA-CD was found to be different under each of the following: uniform traffic, FS traffic with server in most privileged position, FS traffic with server in least privileged position, and RPC traffic.

- *Fairness improves with increasing a under FS and RPC traffic.* In Section 4.2.2 we show that fairness (from the user point of view) for LCSMA-CD under FS traffic with the server in the most privileged position improves with a. The protocol becomes fair for $a \approx 1$. If the server is in the least privileged position, the protocol is fair for small values of a. If a is increased further, most privileged stations from the packet point of view become the least privileged stations from the user point of view. In Section 4.2.4 we show that fairness for LCSMA-CD under RPC traffic improves with increasing a.

- *LCSMA-CD is fair under voice traffic.* Performance under voice traffic is not sensitive to packet delay as long as that delay does not exceed a certain maximum. Therefore, the performance from the user point of view will be equal at all stations as long as the maximum packet delay is not exceeded (more than about 1% of the time) at any station. Therefore, if the traffic load is not too heavy, LCSMA-CD will be fair under voice traffic (Section 4.2.3).

- *Unslotted LCSMA-CD performs better than slotted LCSMA-CD.* The advantage of slotting is that collisions are eliminated. However, bandwidth is wasted since not all slots can be filled entirely. Simulation results show (Section 4.3.1) that the amount of bandwidth saved by eliminating collisions is smaller than the overhead caused by partially filled packets.

- *Fairness depends on the maximum packet length in unslotted protocols.* Long packets have a longer transmission time and therefore are more susceptible to collisions than short packets. Since the number of collisions a packet suffers affects its acquisition time, short packets have lower acquisition times. Simulation results show that fairness

improves if the maximum packet length is limited (Section 4.3.1).

- *Longer service times at the file server improve fairness (from the user point of view).* Since the service time at the server can substantially increase the amount of time a request packet can spend at the server, the acquisition time becomes a smaller component of response time. Therefore, the network becomes less unfair (Section 4.3.1).

- *LCSMA-CD outperforms Expressnet and FASNET under file server traffic.* Simulation results show that LCSMA-CD can outperform Expressnet at transmission speeds of 100 Mbit/s or higher (Section 4.3.1), and FASNET for transmission speeds of 10 Mbit/s or higher (Section 4.3.2).

5.2 Conclusions

- *Fairness should not be measured under uniform one-way traffic.* Measurement of fairness under uniform one–way traffic can be misleading. Results leading to entirely different conclusions can be obtained when fairness is measured under more realistic traffic types.

- *Fairness depends on the point of view.* Fairness depends on the performance measure used to evaluate it. Measurement of fairness from the packet point of view is likely to give different results than measurement of fairness from the user point of view.

- *Unfair protocols (from the packet point of view) often turn out to be fair (from the user point of view).* Unfair protocols, on either the dual or single bus, are fair under request/reply traffic (file server or remote procedure call) for some value of a.

- *Unfair protocols can be better than fair protocols for certain applications.* LCSMA-CD turns out to be better than Expressnet and FASNET for file server traffic. Since unfair (from the packet point of view) protocols turn out to be fair under many conditions, it is possible that unfair protocols will be better than many fair protocols for certain applications.

5.3 Possibilities for Further Research

The following topics may prove to be of interest:

- Comparison of performance of LCSMA-CD to other protocols.
- A list of traffic conditions under which LCSMA-CD is fair, and a formal proof of fairness for each.
- An analytical model for evaluation of fairness under FS and RPC traffic. This model should allow for calculation of a, and the transmission speed, under which LCSMA-CD is fair under FS and RPC traffic.
- Modifications of LCSMA-CD to obtain an upper bound on packet delay for packets transmitted from any station.

APPENDIX A

Bus Topologies and Protocols

This appendix gives descriptions of all the protocols that are mentioned in this document, and the unidirectional bus topologies for which these protocols have been designed.

A.1 Unidirectional Bus Topologies

A.1.1 Single Unidirectional Bus

Figure 22 shows the single unidirectional bus. Stations on the single unidirectional bus transmit on the outbound segment and receive on the inbound segment. In addition, each station has a sense tap on the outbound segment. This tap is used to sense the channel to determine if a transmission from the upstream is in progress.

A.1.2 Dual unidirectional bus.

Figure 23 shows the dual unidirectional bus. If a station has a packet to transmit to a station on the left, then it transmits on the right-to-left (RL) segment. Otherwise, it transmits on the left-to-right segment (LR). In addition to the receive and transmit taps, each station has one sense tap for each bus segment. As on the single unidirectional bus, this tap is used to detect transmissions from the upstream.

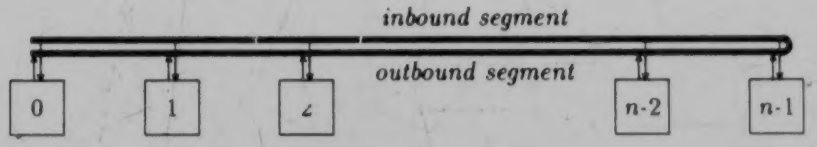


Figure 22: Single unidirectional bus.

Figure 23: Dual unidirectional bus.

A.2 Protocol Descriptions

A.2.1 LCSMA-CD

Local Carrier Sense Multiple Access with Collision Detection (LCSMA-CD) can be used on either the single or dual unidirectional bus. The protocol operates as follows. When a station has a packet to transmit, it first senses the carrier to check if there is a transmission in progress. The sensing is referred to as local because the sensing station cannot tell the status of the carrier at other stations. If the carrier is found to be idle than the station transmits. Otherwise, it waits until the carrier becomes idle. Once the carrier is sensed to be idle the station transmits immediately. If a collision occurs, the station aborts its transmission and allows the packet from the upstream to pass (without damaging it). The station attempts to transmit again when the carrier becomes idle once again.

A.2.2 LCSMA-CD/U/P

The *U* indicates that the protocol is not slotted, whereas the *P* indicates that passive taps are used. Since the protocol is unslotted, packets can be of arbitrary length. Stations connect to the bus through passive taps. The transmit port is an *or* connection, and the receive port simply registers exactly what is being transmitted on the bus. No switching is involved. If active taps were used then a transmitting station could disconnect the portion of the bus carrying the information comming from upstream.

A.2.3 LCSMA-CD/S/P

The *S* indicates slotting, whereas the *P* indicates the use of passive taps. Since the protocol is slotted, the packet length cannot exceed the slot length. If the packet is shorter than the slot length then the whole slot is considered as being used anyway.

A.2.4 LCSMA-CD2

This version of LCSMA-CD is intended for the single unidirectional bus with file server traffic. The protocol is essentially the same as LCSMA-CD/U/P except that the file server behaves a little differently. In particular, the file server does not transmit a reply (even if it has one to transmit) as long it senses activity on its inbound segment.

A.2.5 LCSMA-CDQ

This is a modification of LCSMA-CD/U/P for file server traffic on the single unidirectional bus. The difference between this protocol and LCSMA-CD/U/P is that the file server behaves slightly differently. When the request queue at the server contains more than one request, the requests are sorted such the requests from the less privileged stations are serviced before requests from more privileged stations. The reordering is done each time a new request arrives.

A.2.6 LCSMA-CDQ2

This protocol combines LCSMA-CD2 and LCSMA-CDQ2 into one. The request queue is reordered every time a new request arrives, and also the server waits for silence on its inbound segment before transmitting a reply.

A.2.7 p_i-Persistent

This is LCSMA-CD/S/P with the following modification. Each station is assigned a probability, p_i, where i is the index of the station. If a station has a packet to transmit, and an empty slot is passing by, the station transmits the packet with the probability p_i. These probabilities can be adjusted (possibly dynamically) to make the protocol fair.

A.2.8 DQDB

The Distributed Queue Dual Bus (DQDB) protocol is intended for the slotted dual bus only. The slots carry some extra bits of information, however. Suppose that the two segments

of the bus are labeled A and B. If a station has a packet to transmit on segment A, then it makes a request on segment B by setting an appropriate bit in the passing slot to 1. Each station maintains two request counters RC_A and RC_B; one for each segment. RC_A is incremented whenever a request is registered on segment B. RC_A is decremented (if it is not zero) whenever an empty slot passes the station on segment A (a downstream station will be able to use that slot). A station can transmit on segment A, only when $RC_A = 0$. RC_B is used congruently.

A.2.9 Expressnet

This protocol is unslotted, and intended for the "S" shaped unidirectional bus. Stations receive on the top segment of the "S" and transmit on the bottom segment. The operation of the Expressnet [57] is as follows. A station s, senses the outbound channel, and if the channel is busy it delays its transmission until it senses the end of carrier EOC(OUT). If s detects a transmission from the upstream, it aborts the transmission of its own packet without damaging the packet from upstream. Station s is allowed to transmit again only after detection of end of train on its inbound channel EOT(IN). The bus is "S" shaped so that the stations detect EOT(IN) in the same order as EOC(OUT) is detected. To keep the network alive, stations transmit a burst of carrier, called the "locomotive" wherever EOT(IN) is detected. The burst from different stations overlap.

A.2.10 FASNET

This protocol is slotted, and intended for the dual bus topology [28]. As in DQDB, each slot begins with several bits which may be read as they pass by a station. As in Expressnet, packets are transmitted in trains, and the end-stations are responsible for starting the trains. When one of the end-stations detects the end of train (this is usually an empty slot), then it asserts the start of train bit in the new slot on the other bus segment. Otherwise, stations wait until the start of train is detected, and then transmit when an empty slot passes by.

Bibliography

[1] C. Barberis and D. Pazzaglia. Analysis and optimal design of a packet-voice receiver. *IEEE Tran. Commun.*, COM-28:217–227, Feb. 1980.

[2] B. N. Bershad, T. E. Anderson, E. D. Lazowska, and H. M. Levy. Lightweight remote procedure call. In *Twelfth ACM Symposium on Operating Systems Principles*, pages 102–113, 1989.

[3] T. Bially, B. Gold, and S. Seneff. A technique for adaptive voice flow control in integrated packet networks. *IEEE Tran. Commun.*, COM-28(3), Mar. 1980.

[4] A. D. Birrelll and B. Nelson. Implementing remote procedure calls. *ACM Transactions on Computer Systems*, pages 39–59, Feb. 1984.

[5] P. Brady. A statistical analysis of on-off patterns in 16 conversations. *Bell Syst. Tech. J.*, 47:73–91, Jan. 1968.

[6] P. Brady. A model for generating on-off patterns in two-way communications. *Bell Syst. Tech. J.*, 48:2445–2472, Sep. 1969.

[7] T. M. Chen, J. Walrand, and D. G. Messerschmitt. Dynamic priority protocols for packet voice. *IEEE J. Select. Areas in Commun.*, 7(5):632–643, Jun. 1989.

[8] E. T. Clemmer. Subjective evaluation of transmission delay in telephone conversations. *Bell Syst. Tech. J.*, pages 1141–1147, Jul.-Aug. 1967.

[9] P. Davids, T. Welzel, and T. Anwar. Performance analysis of the dqdb metropolitan area network. In *Singapore International Conference on Networks*, pages 222–227, 1989.

[10] W. Dobosiewicz, P. Gburzynski, and V. Maciejewski. A classification of fairness and performance measures for local area networks and protocols. Submitted for publication.

[11] J. W. Emling. The effect of time delay and echoes on telephone conversations. *Bell Syst. Tech. J.*, 51(7), Jan. 1963.

[12] J. Filipiak. Access protection for fairness in a distributed queue dual bus metropolitan area network. In *IEEE International Conference on Communications '89*, pages 635–9, 1989.

[13] M. Fine and F.A. Tobagi. Demand assignment multiple access schemes in broadcast bus local area networks. *IEEE Transactions on Computers*, 33(12):1130–1159, dec 1984.

[14] M. J. Fisher. Delay analysis of tasi with random fluctuations in the number of voice calls. *IEEE Tran. Commun.*, COM-28:1883–1889, Nov. 1984.

[15] E. Fuchs and P. Jackson. Estimates of distributions of random variables for certain computer communications traffic models. *Communications of the ACM*, 13(12):752–757, Dec. 1970.

[16] P. Gburzyński and P. Rudnicki. *The LANSF Protocol Modeling Environment, version 2.0.* The University of Alberta, Department of Computing Science, TR 891 , Edmonton, 1989.

[17] J. G. Gruber and N. H. Le. Performance requirements for integrated voice data networks. *IEEE J. Select. Areas in Commun.*, SAC-1:981–1005, Dec. 1983.

[18] J. G. Gruber and L. Stawczynski. Subjective effects of variable delay and speech clipping in dynamically managed voice systems. *IEEE Tran. Commun.*, COM-33(8):801–808, Aug. 1985.

[19] Haskel. Buffer and channel sharing by several interframe picturephone coders. *Bell Syst. Tech. J.*, 51(1):261–289, Jan. 1972.

[20] B. G. Haskell, F. W. Mounts, and J. C. Candy. Interframe coding of videotelephone pictures. *Proc. of the IEEE*, 60:792–800, Jul. 1972.

[21] J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. West. Scale and performance in a distributed file system. *ACM Trans. on Computer Systems*, 6(1):51–81, Feb. 1988.

[22] P.E. Jackson and C.D. Stubbs. A study of multiaccess computer communications. In *AFIPS 1969 SJCC*, volume 34, pages 491–504, 1969.

[23] G. Karlsson and M. Vetterli. Packet video and its integration into the network architecture. *IEEE J. Select. Areas in Commun.*, 7(5):739–751, Jun. 1989.

[24] S. Kasemlonnapa, J.S. Meditch, and B. Mukherjee. A delay-throughput performance analysis of the p_i-persistent protocol for unidirectional broadcast bus networks. In *IEEE INFOCOM'89 The Conference on Computer Communications*, volume 3, pages 834–40, 1989.

[25] T. Koga, Y. Iijima, I. Kazumoto, and T. Ishiguro. Statistical performance analysis of an interframe encoder for broadcast television signals. *IEEE Tran. Commun.*, COM-29(12):1868–1875, Dec. 1981.

[26] Bell Telephone Labs. *Engineering and Operations in the Bell System.* AT and T Bell Labs, 1984.

[27] E. D. Lazowska, J. Zahorjan, D. R. Cheriton, and W. Zwaenpoel. File access performance of diskless workstations. *ACM Transactions on Computer Systems*, 4(3):238–268, Aug. 1986.

[28] J. O. Limb and L. E. Flamm. A distributed local area network packet protocol for combined voice and data transmission. *IEEE J. Select. Areas in Commun.*, SAC-1(5):926–934, Nov. 1983.

[29] V. Maciejewski, W. Dobosiewicz, and P. Gburzynski. Behavior of unfair protocols under file server traffic. Submitted for publication.

[30] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Tran. Commun.*, 36:834–844, Jul. 1988.

[31] M. A. Marsan and M. Gerla. Fairness in local computer networks. In *IEEE International Conference on Communications '82*, volume 1, pages 2F.4/1–6, 1982.

[32] W. T. Marshal and S. P. Morgan. Statistics of mixed data traffic on a local area network. *Computer Networks and ISDN systems*, pages 185–194, Oct., Nov. 1985.

[33] N. F. Maxemchuk. Twelve random access strategies for fiber optic networks. *IEEE Tran. Commun.*, 36(8):924–950, Aug. 1988.

[34] B. W. Meister, P. A. Janson, and L. Svobodova. Connection-oriented versus connectionless protocols: A performance study. *IEEE Tran. Comp.*, C-34(12):1164–1173, Dec. 1985.

[35] D. Minoli. Optimal packet length for packet voice communication. *IEEE Tran. Commun.*, COM-27(3):607–611, Mar. 1979.

[36] B. Mukherjee. On the infinite buffer model and the implementation aspects of the p_i-persistent protocol for unidirectional broadcast bus networks. In *IEEE International Conference on Communications '88*, volume 1, pages 273–7, 1988.

[37] B. Mukherjee and J.S. Meditch. The p_i-persistent protocol for unidirectional broadcast bus networks. In *IEEE International Conference on Communications '87*, volume 1, pages 563–7, 1987.

[38] B. Mukherjee and J.S. Meditch. The p_i-persistent protocols for unidirectional broadcast bus networks. *IEEE Tran. Commun.*, 36(12):1277–86, Dec. 1988.

[39] S. J. Mullender and A. S. Tanenbaum. Immediate files. *Software Practice and Experience*, 14(4):365–368, Apr. 1984.

[40] H. Murakami, S. Matsumoto, Y. Hatori, and H. Yamamoto. 15/30 mbits/s universal digital tv codec using a median adaptive predictive coding method. *IEEE Tran. Commun.*, COM-35:637–645, Jun. 1987.

[41] A. N. Netravali and B. G. Haskell. *Digital Pictures: Representation and Compression.* New York: Plenum, 1989.

[42] A. N. Netravali and J. O. Limb. Picture coding: A review. *Proc. of the IEEE*, 68(3):366–406, Mar. 1980.

[43] M. Nomura, T. Fujii, and N. Ohta. Basic characteristics of variable rate video coding in atm environment. *IEEE J. Select. Areas in Commun.*, 7(5):752–760, Jun. 1989.

[44] J. K. Ousterhout, H. Da Costa, D. Harrison, J. A. Kunze, M. Kupfer, and J. G. Thompson. A trace-driven analysis of the unix 4.2 bsd file system. In *The Tenth ACM Symposium on Operating Systems Principles*, pages 15–24, Dec. 1985.

[45] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw Hill, 1984.

[46] P. F. Pawlita. Traffic measurement in data networks, recent measurements and results and some implications. *IEEE Tran. Commun.*, COM-29(4):525–535, April 1981.

[47] P. Rudnicki. Personal Communication.

[48] S. Sabri and B. Prasada. Videoconferencig systems. *Proc. of the IEEE*, 73(4):671–688, Apr. 1985.

[49] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation of the sun network filesystem. In *Useniz Conference and Exhibition*, pages 119–130, Summer 1985.

[50] M. D. Schroeder and M. Burrows. Performance of firefly rpc. In *Twelfth ACM Symposium on Operating Systems Principles*, pages 83–90, 1989.

[51] P. Sen, B. Maglaris, Nasser-Eddine Rikli, and D. Anastassiou. Models for packet switching of variable-bit-rate video sources. *IEEE J. Select. Areas in Commun.*, 7(5):865–869, Jun. 1989.

[52] Shan shan Huang. Source modeling for packet video. In *IEEE ICC'88*, pages 1267–1266, 1988.

[53] W. Stallings. *Local Networks: An Introduction.* Macmillan, 1987.

[54] T. Suda, H. Miyahara, and T. Hagesawa. Performance evaluation of a packetized voice system – simulation study. *IEEE Tran. Commun.*, COM-32(1):97–102, Jan. 1984.

[55] L. Svobodova. *Computer Performance Measurement and Evaluation Methods: Analysis and Applications.* Elsevier, 1976.

[56] L. Svobodova. File servers for network-based distributed systems. *Computing Surveys*, 16(4):353–398, Dec. 1984.

[57] F. A. Tobagi, F. Borgonovo, and L. Frata. Expressnet: A high-performance integrated-services local area network. *IEEE J. Select. Areas in Commun.*, SAC-1(5):898–913, Nov. 1983.

[58] R. van Renesse, A. S. Tanenbaum, and A. Wilschut. The design of a high-performance file server. In *The 9th International Conference on Distributed Computing Systems*, pages 22–27, Jun. 5-9 1989.

[59] R. van Renesse, H. van Staveren, and A. S. Tanenbaum. The performance of the amoeba distributed operating system. *Software Practice and Experience*, 19(3):223–234, Mar. 1989.

[60] W. Verbiest, L. Pinnoo, and B. Voeten. The impact of the adt concept on video coding. *IEEE J. Select. Areas in Commun.*, 6:1623–1632, Dec. 1988.

[61] C. J. Weinstein and J. W. Forgie. Experience with speech communication in packet networks. *IEEE J. Select. Areas in Commun.*, SAC-1(6):963–980, Dec. 1983.

[62] H. Yamamoto, Y. Hatori, and H. Murakami. 30 mbit/s codec for ntsc color tv signal using interfield–intrafield adaptive prediction. *IEEE Tran. Commun.*, COM-29:1859–1867, Dec. 1981.